Development and Interpretation of Machine Learning Models for Drug Discovery

Kumulative Dissertation

zur Erlangung des Doktorgrades (Dr. rer. nat.)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Jenny Balfer

aus Bergisch Gladbach

Bonn 2015

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Jürgen Bajorath
2. Gutachter: Prof. Dr. Andreas Weber
Tag der Promotion: 22. Oktober 2015
Erscheinungsjahr: 2015

Abstract

In drug discovery, domain experts from different fields such as medicinal chemistry, biology, and computer science often collaborate to develop novel pharmaceutical agents. Computational models developed in this process must be correct and reliable, but at the same time interpretable. Their findings have to be accessible by experts from other fields than computer science to validate and improve them with domain knowledge. Only if this is the case, the interdisciplinary teams are able to communicate their scientific results both precisely and intuitively.

This work is concerned with the development and interpretation of machine learning models for drug discovery. To this end, it describes the design and application of computational models for specialized use cases, such as compound profiling and hit expansion. Novel insights into machine learning for ligand-based virtual screening are presented, and limitations in the modeling of compound potency values are highlighted. It is shown that compound activity can be predicted based on high-dimensional target profiles, without the presence of molecular structures. Moreover, support vector regression for potency prediction is carefully analyzed, and a systematic misprediction of highly potent ligands is discovered.

Furthermore, a key aspect is the interpretation and chemically accessible representation of the models. Therefore, this thesis focuses especially on methods to better understand and communicate modeling results. To this end, two interactive visualizations for the assessment of naïve Bayes and support vector machine models on molecular fingerprints are presented. These visual representations of virtual screening models are designed to provide an intuitive chemical interpretation of the results.

Acknowledgements

I would like to thank my supervisor Prof. Dr. Jürgen Bajorath for providing a work environment in which I could pursue my own ideas at any time, and for all his motivation and support. Furthermore, thanks go to Prof. Dr. Andreas Weber, who agreed to be the co-referent of this thesis, and the other members of my PhD committee. Dr. Jens Behley, Norbert Furtmann, and Antonio de la Vega de León improved this thesis by many valuable comments and suggestions.

I am also grateful to my colleagues from the LSI department, who created a friendly team environment at any time. Especially, Dr. Kathrin Heikamp gave me many advices and cheered me up on countless occasions. Norbert Furtmann agreed to show me real lab work and was a great programming student. Antonio de la Vega de León was my autumn jogging partner and endured all my lessons about the Rheinland culture, and Disha Gupta-Ostermann was a very nice office neighbor (a.k.a. stapler girl).

My deepest gratitude goes to Jens Behley, without whom I would have never started, let alone finished my PhD thesis. His constant and ongoing support is invaluable.

Finally, I would like to dedicate this work to the memory of Anna-Maria Pickard, Wilhelm Balfer, and Sven Behley.

Contents

Acronyms

2D	two-dimensional.
3D	three-dimensional.
ADME	absorption, distribution, metabolism and excretion.
ANN	artificial neural network.
ECFP4	extended connectivity fingerprint with bond diameter 4.
GPCR	G-protein coupled receptor.
HTS	high-throughput screening.
KKT	Karush-Kuhn-Tucker.
LASSO	layered skeleton-scaffold organization.
LBVS	ligand-based virtual screening.
MACCS	molecular access system.
MMP	matched molecular pair.
MMS	matching molecular series.
MOE	molecular operating environment.
NSG	network-like similarity graph.
SAR	structure-activity relationship.
SARI	SAR index.
SAS	structure-activity similarity.
SBVS	structure-based virtual screening.
SMARTS	SMILES arbitrary target specification.
SMILES	simplified molecular-input line entry system.
SVM	support vector machine.
SVR	support vector regression.
Tc	Tanimoto coefficient.
TGT	typed graph triangles.

Introduction

1 Motivation

In the past century, the systematic discovery and development of drugs has tremendously changed our ability to treat diseases. While until the late 19th century, only naturally occurring drugs were known, the advent of molecular synthesis disclosed a whole new field of research [1, 2]. Since then, the field of drug development has evolved rapidly, enabling the treatment of formerly immedicable conditions such as syphilis or polio. However, the progress of finding a drug to treat a certain disease is a complicated, expensive, and time-consuming process: a recent study estimates the cost for the development of one new drug at US $2.6 billion [3, 4].

Today, computational or *in silico* modeling is applied during many steps of the drug development process. In contrast to *in vitro* testing, i.e., the generation of experimental data in a laboratory, computer-based methods are comparably fast and cheap. However, *in silico* models are far from perfect and can as such only complement and never substitute *in vitro* modeling. Nevertheless, they are important tools for pre-screening compound libraries or, maybe even more importantly, for understanding certain chemical phenomena. Here, the idea is to use elements from the field of machine learning and pattern extraction to explain observed aspects of medicinal chemistry.

The main focus of this thesis is the development and interpretation of machine learning models for pharmaceutical tasks. In drug discovery, project teams usually consist of experts from a variety of disciplines, including biology, chemistry, pharmacy, and computer science. *In silico* models therefore do not only need to be as accurate as possible and numerically interpretable to the computer scientist, but also chemically interpretable to the experts from the life sciences. This thesis focuses on the understanding of computational models for drug discovery, and introduces chemically intuitive interpretations. Thereby, we hope to contribute to further enhanced communication in interdisciplinary drug development teams.

2 The drug development process

Drug development describes the process of developing a pharmaceutical agent to treat a certain disease. This process can be divided into five major steps (cf. figure 1): (1) Target selection, (2) hit compound identification, (3) hit-to-lead optimization, (4) preclinical and (5) clinical drug development.

Target identification aims to find a biological target that can be activated or inhibited to prevent or cure the disease. This can be, for example, an ion channel, a receptor, or an enzyme. Popular drug targets include G-protein coupled receptors (GPCRs) or protein kinases [5, 6]. Once a target is identified, one searches for a so-called hit compound. This is a small molecule that has an activity against the target, but lacks other characteristics important for the final drug. For example, the hit compound may only have intermediate potency, lack specificity, or be toxic. In order to find a hit compound, a large library of molecules has to be screened against the target. This can be either modeled computationally or done *in vitro* by high-throughput screening (HTS).

After one or more hit compounds are identified, they are subjected to hit-to-lead optimization. The hits are optimized by exchanging functional groups to obtain ligands that are also active against the target, but act more potent, display less side effects, or have other preferred characteristics. Important parameters are for instance the absorption, distribution, metabolism and excretion (ADME) properties that describe how a drug behaves in the human body. To optimize these parameters for "drug-likeliness", Lipinski and colleagues introduced their famous "rule of five" that ligands should obey, including for example a molecular weight below 500 Da or at most five hydrogen bond donors [7, 8].

From the ligands that are obtained from hit-to-lead optimization, one or more lead compounds are chosen. These are then subjected to preclinical research, which includes further *in vitro* and first *in vivo* tests. The major goal of the preclinical stage is to

Figure 1: The major steps of the drug development process.

determine whether it is safe to test the drug in clinical trials, where the drug is tested in a group of different individuals to finally evaluate how it interacts with the human organism.

If all these stages have successfully been passed, the drug can be submitted to the responsible administration facility. Passing all stages of drug development takes several years, and failures become more expensive the later they occur in the process. Thus, it is desirable to optimize the earlier stages of drug development, so that only the most promising compounds will enter the expensive preclinical and clinical trials.

Computational modeling is applied in the first three states of the drug development process, which form the task of drug discovery. In this context, one also often speaks of chemoinformatics. Disease pathways are modeled and analyzed in order to identify targets. Furthermore, computational approaches for the design of maximally diverse and promising compound libraries are applied in the hit identification stage. If the crystal structure of the target is known and its binding sites are identified, docking can be applied to find active hits. Docking is a type of structure-based virtual screening (SBVS), where one tries to find ligand conformations that best fit into the binding pocket of the target.

In contrast, the main theme of this thesis is ligand-based virtual screening (LBVS). Here, the idea is to extrapolate from ligands with known activity to previously untested ones. As such, it is applicable in the lead optimization stage, when at least one active compound has been identified. LBVS studies covered in this thesis include the prediction of compound activity, the modeling of potency values, and the profiling of ligands against a panel of related targets.

Aside from the development of LBVS methods, understanding the resulting models is a key aspect in drug discovery. Beneath the correct identification of active or highly potent ligands, it is crucial to understand what features of the compounds determine the desired effect. These results then need to be communicated to the pharmaceutical experts to validate or improve the models using domain knowledge. An intuitive explanation of a model's decision can also help to better understand the structure-activity relationship of the ligand-target complex, aid in the improvement of the model itself, and is of great importance for communication in an interdisciplinary team. Furthermore, interpreting an LBVS model can provide a ligand-centric view on the characteristics that determine biological activity. This is opposed to the target-centric view that structure-based modeling provides, and is especially important when the target's crystal structure is unknown.

In this thesis, both the development and the interpretation of machine learning for LBVS will be covered. Hence, the following chapter will introduce some basic concepts of *in silico* modeling for drug discovery.

3 Concepts

Machine learning models for drug discovery mostly try to model the structure-activity relationship of ligand-target interactions. To build a predictive model, several components are required: (a) molecular data in a suitable representation, (b) a similarity metric that quantitatively compares two molecules (depending on the algorithm), and (c) a learning algorithm to compute the parameters of the final model. This chapter will first introduce the concept of structure-activity relationship. Then, small molecule data sources and possible representations are discussed. Next, common similarity metrics and learning algorithms are introduced.

3.1 Structure-activity relationship

While there are efforts to model the physicochemical properties of ligands [9–11] or predict drug-likeliness [12, 13], most LBVS approaches aim to model the structure-activity relationship (SAR) of ligands [14]. As the name suggests, structure-activity relationship (SAR) analysis aims to explain the relationship between a compound's chemical structure and its activity against a certain target. SAR modeling approaches are usually based on the similarity property principle, which states that compounds with similar structure should exhibit similar properties [15]. Hence, most models try to extrapolate from the activity of known ligands to the activity of structurally similar ones. However, in LBVS one is usually interested in recovering new active ligands that are distinct from the known ones to a certain extent [16]. This is because for the discovery of close analogs, a complex machine learning algorithm is not required. Hence, the goal is to identify ligands that are similar enough to the known actives to share their activity, but distinct enough to expand to new regions of the chemical space.

If the similarity property principle holds and similar structures share similar activities, one also speaks of *continuous SAR*. Contrary, the term *discontinuous SAR* is used if similar structures exhibit large differences in their potencies [17]. SAR continuity and discontinuity can be expressed both locally and globally, quantitatively by scores such as the SAR index (SARI) [18], or qualitatively through visualization techniques. An extreme form of SAR discontinuity are so-called *activity cliffs*, pairs of similar ligands with a large potency difference [19]. Despite the known fact that SAR continuity and discontinuity strongly depends on the chosen molecular representation and similarity measure, activity cliffs are believed to be focal points of SAR analysis and therefore widely studied [20–23].

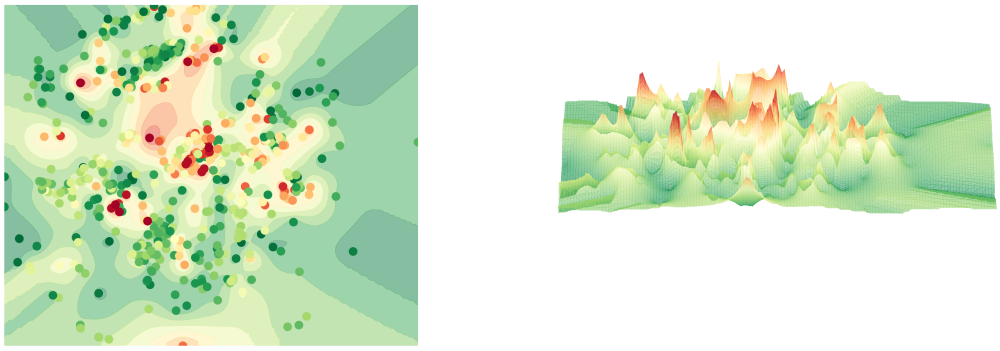


Figure 2: Exemplary 2D and 3D SAR landscapes for a set of human thrombin ligands.

SARs are often studied qualitatively in visual form. Therefore, a number of visualization methods has been developed focusing on different SAR characteristics [24, 25]. The probably most intuitive visualizations include two-dimensional (2D) and three-dimensional (3D) SAR landscapes [26]. Here, the compounds are projected into 2D space by a similarity-preserving mapping, for example derived by multidimensional scaling [27]. Then, they are augmented by their potency annotations, which are visualized by coloring (2D landscapes) or as coordinates on a third axis (3D landscapes). The advantage of these visualizations is that continuous and discontinuous SAR can be intuitively accessed, as can be seen from figure 2. A variety of other visualizations have been developed, including network-like similarity graphs (NSGs) [28], layered skeleton-scaffold organization (LASSO) graphs [29], or structure-activity similarity (SAS) maps [30].

In chapter 4, both quantitative and qualitative measures of SAR continuity are used to provide a critical view on potency modeling using support vector regression.

3.2 Molecule data sources and potency measurements

Typically, ligands are small organic molecules with a molecular weight lower than 500 Da [31]. Millions of structures are available in publicly accessible compound databases, and even more in proprietary portfolios. Some of the largest public databases are ZINC [32], PubChem [33, 34], and ChEMBL [35].

ZINC contains the 3D structures of over 35 million commercially available compounds. Furthermore, subsets of lead-like, fragment-like, and drug-like compounds are provided, as well as shards. PubChem is split into three main databases: PubChem Substance, Compound, and BioAssay. While the Substance database contains all chemical names and structures submitted to PubChem, the PubChem Compound database contains only unique and validated compounds. The BioAssay depository contains descriptions of assays and the associated bioactivity data, which are linked to the other two databases.

As of April 2015, PubChem contains over 68 million compounds, of which roughly 2 million were tested in 1.15 million bioactivity assays, leading to more than 220 million activity annotations. ChEMBL contains more than 13.5 million activities of roughly 1.7 million compounds against 10,000 targets (version 20). It is a collection of manually curated data from primary published literature and updated regularly.

In some parts of this thesis, compounds are either classified as active or inactive, depending on whether the strength of their interaction with the target exceeds a certain threshold. Other chapters use their potency values for regression analysis. The way these potencies are measured however depends on the data source and the information provided.

In chapter 1 and chapter 3, percentages of residual kinase activity at a given compound concentration are utilized. Here, the activity of a kinase is first measured in absence of the compound to be tested, and the obtained value is set to 100 %. Then, the compound is added at a defined concentration. If it inhibits the kinase activity, only a reduced value of activity will be measured: this is the relative residual activity. The compounds used in chapter 3 were also tested for their residual activity. Furthermore, for all compounds that inhibited a kinase to less than 35 % of its original activity, a K_d value was determined. The K_d value is the thermodynamic dissociation constant. The lower this concentration, the higher is the binding affinity, or potency, of the compound.

In chapter 4, the ligands considered for modeling are required to have a K_i value below 100 µM. K_i values are absolute inhibition constants, which can be used to compare potencies across assays with different conditions. They can be determined from half-maximal inhibitory concentrations (IC_{50} values). In contrast to the K_d values used in chapter 1 and chapter 3, IC_{50} values are not determined at a single compound concentration. Instead, a dose-response curve is generated at different compound concentrations, and the concentration is determined at which half-maximal inhibition is reached. Since the IC_{50} value depends on the assay conditions, i.e., it can be influenced by the enzyme or substrate concentrations, it can be converted into a K_i value [36, 37]. Here, assay concentrations are considered and the values are hence comparable across different assays.

Besides K_d, K_i, or IC_{50} values, literature often reports logarithmically transformed pK_d, pK_i, or pIC_{50} values. Here, one calculates the negative logarithm of the original potency value in molar, i.e., $pK_i = -\log_{10}(K_i)$. This scale is usually seen as more intuitive, since higher values indicate stronger binding affinity. Furthermore, negative logarithmic values remain interpretable in the sense that each integer corresponds to one order of magnitude, i.e., a value of 6 pK_i corresponds to 1 µM K_i, while a value of 9 pK_i corresponds to 1 nM K_i.

3.3 Data representation

Small molecules are most naturally represented as graphs, where each node corresponds to an atom and each edge to a bond. 2D molecular graphs can be easily visualized on screen and paper, and are intuitively comprehensible by medicinal chemists.

However, molecular graph representations for computational screening have the disadvantage that they require a lot of digital resources compared to other representations. First, all graph nodes and edges have to be stored, and second, graph comparisons are computationally expensive. Therefore, many digital representations have been developed that require less computational resources. Probably the most popular example for a digital molecular representation are simplified molecular-input line entry system (SMILES) strings [38–41]. SMILES encode the molecular graph as a linear ASCII string. The elemental symbol of each atom is used, and single bonds are omitted between neighboring atoms. Parentheses denote branching, and there are special symbols for aromaticity, stereochemistry, or isotopes. Furthermore, an extension called SMILES arbitrary target specification (SMARTS) has been developed that allows the use of wild cards and patterns for database queries.

While SMILES strings are suitable for storing large amounts of molecules with minimal storage requirements, they still have to be converted back to a molecular graph to work with them. However, for fast similarity assessment, it is reasonable to describe ligands not by their structure, but by certain features. For this purpose, molecules are often represented as vectors of real-valued descriptors, or as molecular fingerprints. A large variety of molecular descriptors exist, from simple atom counts or defined values like the molecular weight or water solubility of a compound to more complex ones, such as shape indices [42, 43]. Several of these descriptors together in a vector can serve as an abstract, yet discriminative description of a molecule. They are numerically accessible and can be compared in fast and clearly defined ways.

A prominent case of numerical compound descriptions are molecular fingerprints. These are bit vectors where each position is set to 1 or 0, depending on whether a certain feature is present or absent in the given molecule. A variety of molecular fingerprints have been developed. The most common ones can be divided into substructural, pharmacophore, and extended connectivity fingerprints. Substructural fingerprints are fixed-length sets of pre-defined substructures, where each substructure is associated with a certain position in the bit string. To encode a molecule, the bit positions of all substructures that are present are set to 1, while the other positions are set to 0. One of the most popular substructural fingerprint are molecular access system (MACCS) keys, which consist of 166 pre-defined substructures [44]. Pharmacophore fingerprints usually proceed by assigning each atom one pre-defined type, for instance “hydrogen donor” (D), “hydrogen acceptor” (A), or “hydrophobic” (H). Then, all sets of atoms of a certain length are encoded using the graph distances between the sets’ members and their atom types. Common pharmacophore fingerprints implemented in the molecular operating environment (MOE) are GpiDAPH3, typed graph triangles (TGT), or piDAPH4, which encode pairs, triplets, or quadruplets of atoms, respectively [45]. Extended connectivity fingerprints are a class of topological fingerprints, where for each

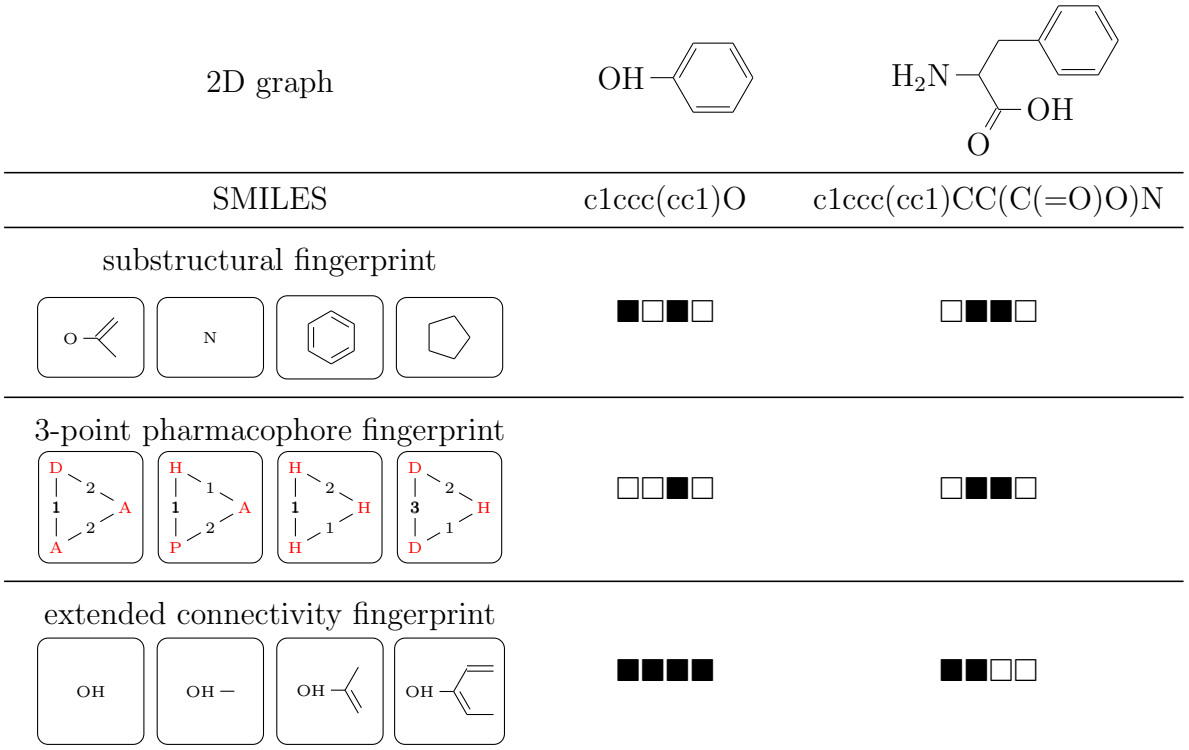


Figure 3: Molecular graphs of phenol and phenylalanine, their SMILES representations, and schematic visualization of the MACCS, TGT, and ECFP4 fingerprints. Black squares indicate set bits, ie., present structures, whereas white squares represent bits that are set to 0.

atom, its circular environment up to a specific bond length is enumerated [46]. Then, each unique environment is mapped to a number using a hash function. By design, extended connectivity fingerprints do not have a fixed length. Instead, the number of bits is variable and depends on the data set. Figure 3 schematically compares a substructural, pharmacophore, and extended connectivity fingerprint with four bits each on the example of two small molecules.

Throughout this thesis, MACCS and the extended connectivity fingerprint with bond diameter 4 (ECFP4) are used to represent ligands. Both can be computed from the 2D molecular graph and do not require a known 3D conformation. Additionally, matched molecular pairs and activity-based fingerprints are used in chapter 2 and chapter 3, respectively. The decision to use fingerprints over real-valued descriptor vectors is motivated by two reasons. First, calculations on binary fingerprints are fast and not prone to floating point errors. Second, it is possible to back-project any set feature back onto the molecular graph and hence provide a visual explanation of each fingerprint. Thereby, molecular fingerprints are more easily interpretable than value ranges of other descriptors. We will exploit this especially in part III of this thesis.

The specific fingerprints MACCS and ECFP4 were chosen because they represent two separate classes of fingerprints with very different complexity. While MACCS has a

fixed length of 166 bits, each encoding a specifically predefined substructure, ECFP4 is of variable length and the substructures encoded by each bit depend on the data sets. Furthermore, their typical similarity value distributions across data sets show different characteristics: while MACCS usually produces broad normal distributions of Tanimoto coefficient values centered around 0.4 to 0.6, the Tanimoto coefficient distributions of ECFP4 are not normally distributed, have small standard deviations and a mean below 0.25 [47].

3.4 Similarity assessment

Many learning algorithms require a similarity assessment to quantitatively compare two compounds. Several methods exist to derive ligand similarity, depending on the chosen molecular representation. If molecules are represented by graphs, subgraph isomorphisms or graph assignments can be used to determine their similarity. However, the computation of graph kernels is computationally inefficient, since the subgraph isomorphism problem is NP hard [48]. Nevertheless, several similarity metrics for graphs have been introduced, e.g., based on labeled pairs of graph walks [48, 49].

Another popular formalism of similarity for chemical structures is the concept of matched molecular pairs (MMPs). An MMP is defined as a pair of compounds that share a common core and only differ in a limited number of substructures [50] (cf. figure 4). Usually, MMPs are size-restricted, which means that the common core is required to have a minimum size, while the different substructures can only have a maximum number of heavy atoms. Furthermore, the number of exchangable substructures is limited: often, only one substructure is allowed to differ in an MMP. While the MMP formalism induces a rather strict measure of similarity (either a pair of ligands forms an MMP or not), it has the advantage that it is extremely intuitive. Furthermore, the exchanged substructures can often directly be translated to synthesis rules.

In the case of molecular descriptor vectors or fingerprints, similarity can be determined straightforward by existing metrics. Common metrics are for instance the Euclidean, cosine, or cityblock distance. For fingerprints, the Tanimoto similarity [51] has become particularly popular [52]. In this thesis, it is often used as a support vector machine (SVM) kernel.

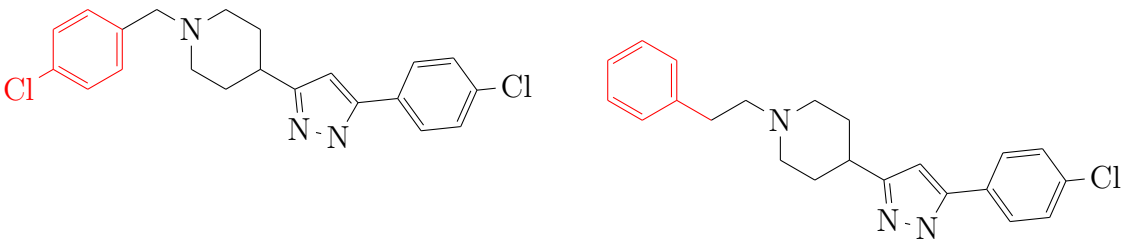


Figure 4: Example for an MMP. The common core is depicted black, while the exchanged substructure is highlighted in red.

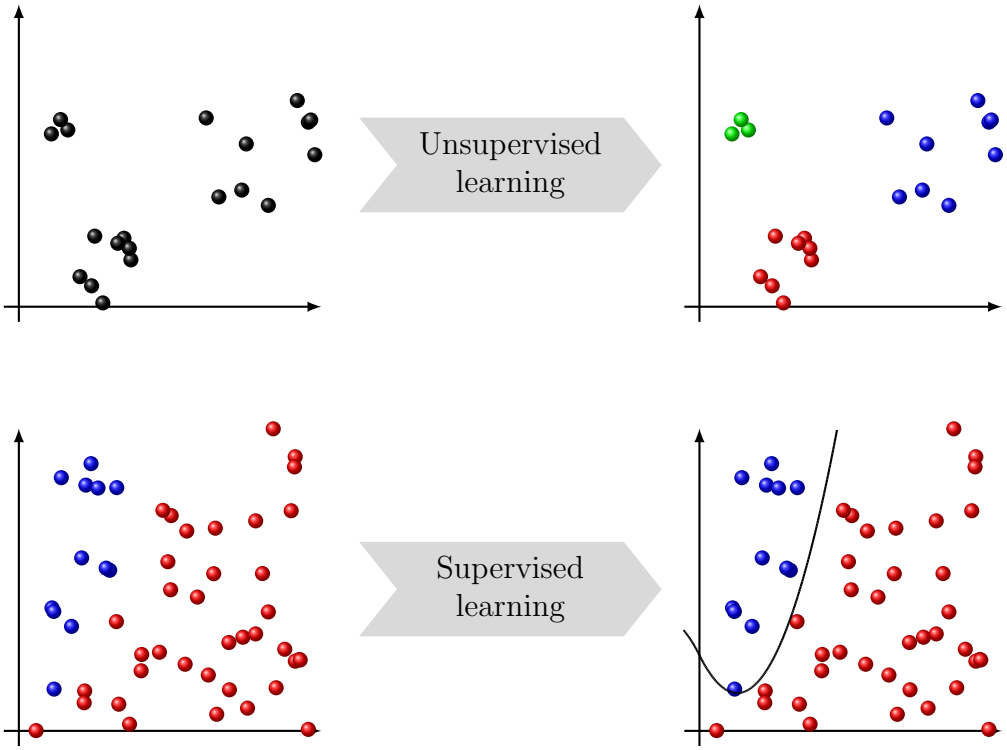


Figure 5: Schematic visualization of unsupervised and supervised learning algorithms.

3.5 Learning algorithms

The final ingredient for a virtual screening model is the learning algorithm. Here, one can distinguish between unsupervised and supervised methods. Unsupervised learning means that the algorithm is given a number of molecules, and aims to detect hidden structure in the data. This can mean to derive groups or clusters of compounds that belong together, or to find and reduce correlated dimensions. In contrast, supervised learning algorithms take a number of molecules and their corresponding labels as input. From both together, they derive a model that is able to predict the label of new, previously unseen instances. Figure 5 schematically illustrates both types of learning. If all possible supervised labels belong to a finite set, the prediction process is called *classification*, whereas one speaks of *regression* in the case of continuous values.

For the purpose of LBVS, one typically employs supervised learning. Here, a set of tested ligands are augmented with their labels, which are often categorical activity annotations (i.e., "active" vs. "inactive") or continuous potency values. The learning algorithm is then supplied with these compounds and labels as the training set. From the training set, the model is derived, which can then be used to predict labels for new and untested compounds. The set of compounds that are previously unknown and used for prediction is called the test set.

Many supervised learning algorithms however do not only require a training set of inputs and labels, but also a number of *hyperparameters*. These parameters have to be

set prior to modeling, as opposed to the *model parameters* that are determined by the respective algorithm. Example for hyperparameters are the choice of k for k-nearest neighbors, the kernel of an SVM, or the number of trees in a random forest. While there may be cases where the choice of hyperparameter values can be determined from the nature of the data or the problem, hyperparameter selection is non-trivial in most settings. Here, one usually employs *cross-validation* to determine the best parameter choices from a set of pre-selected ranges. First, the training data is split into a number of k equally sized folds (hence, one also speaks of k-fold cross-validation). Then, for each hyperparameter choice, the learning algorithm is run k times using the data from $(k-1)$ folds as a training set, and the remaining fold as the validation set. The data from the validation set is unknown to the learning algorithm, and the resulting model is used to predict the labels of this set. Then, an evaluation metric is used to assess the performance of the model on the validation set. This process is repeated for all k folds, and the average performance on the validation sets is used as an indicator of how well the current hyperparameters perform on the given data. Figure 6 visualizes this approach on the example of a learning algorithm that fits a polynomial to classify the data. Here, the order of the polynomial has to be given as a hyperparameter, and polynomials of the first, second, and third order are validated.

While it is generally possible to use k equal to the number of training compounds, and hence produce a so-called *leave-one-out* estimate of hyperparameter performance, k is often chosen to be 5 or 10 in practice. In fact, there are studies recommending to use 10-fold over n-fold cross validation [53]. Using a limited number of folds also reduces the time complexity of the cross-validation, which can be an important factor especially when several hyperparameters with large ranges have to be evaluated.

The most commonly applied learning algorithms in chemoinformatics include artificial neural networks (ANNs), decision trees and random forests, SVMs, k nearest neighbors, and naïve Bayes [52, 54]. ANNs use layers of single perceptron units, inspired by the network of neurons in the human brain [55]. Usually, there is one layer of artificial input neurons, one layer of output neurons, and a number of neurons organized in one or more hidden neuron layers in between. All layers are interconnected, and the algorithm proceeds by learning the weights of the neurons' functions. While multi-layered ANNs can be extremely powerful, they are also hard to interpret, especially when the number of hidden layers and units grows [56].

Decision trees derive a set of rules from the training data, which can then be used to classify the test data [57]. Here, the training data is recursively split into subsets by the descriptor that best separates the remaining data. Overall, this recursive procedure creates a tree of *if-then-else* decision rules. Single decision tree models are therefore easily interpretable, yet can be prone to overfitting [58]. Hence, ensemble classifiers using multiple trees have been developed, the so-called random forests [59]. Here, several trees are grown and then combined by a voting procedure to arrive at a final classification.

SVMs are classifiers developed for the separation of two different classes [60]. The idea is to fit a plane in high-dimensional space through the training data, and classify the test data based on the side of the hyperplane they fall. Since SVM models are used extensively in this thesis, they will be discussed in detail in the following chapter.

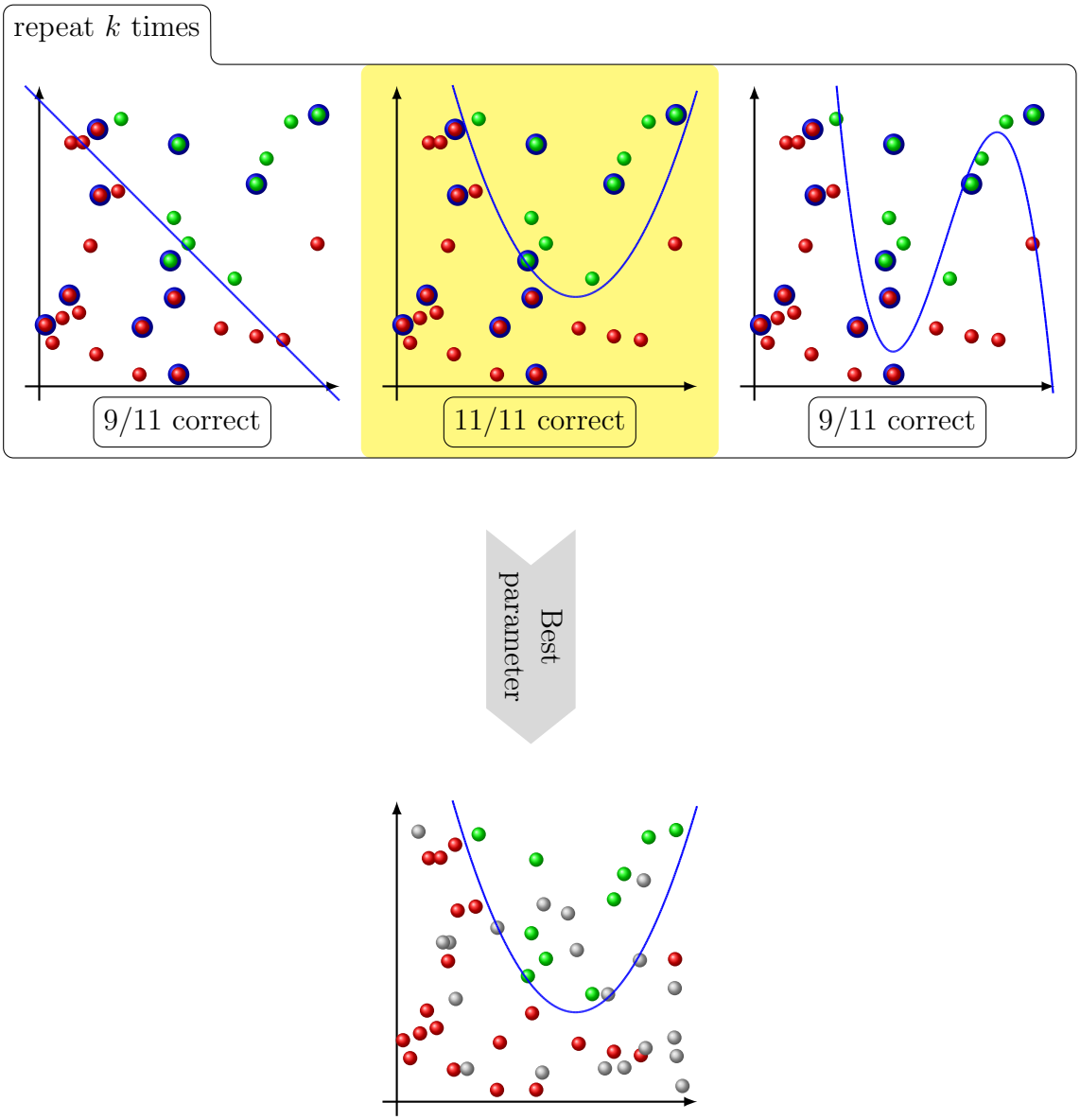


Figure 6: Schematic depiction of cross-validation for hyperparameter selection. The training data is divided k times into a training set (red and green circles) and a validation set (circles with blue border). Each hyperparameter is once used to build a model, and the number of correctly predicted compounds from the validation set is reported. The parameter that gives the best performance, here the polynomial of degree 2, is chosen to build the final model on the complete training set. This final model is then used to predict the classes of the test instances (gray circles).

The k nearest neighbor algorithm is one of the simplest classifiers and often used for chemical similarity searching [61]. Here, one calculates the distances of the test compounds to each training compound. The class label of the k nearest neighbors is then chosen as the prediction for the test compounds. This approach can also be applied if only one class, for instance active ligands, are given. Test compounds are then ranked by their average similarity to the k nearest neighbors of the unlabeled training set. While k nearest neighbor classification is simple and interpretable, it is computationally expensive due to the pairwise distance calculations, and often less powerful than more sophisticated learning algorithms [62].

Naïve Bayes classifiers are generative models that use Bayes' theorem to predict the probability of each test instance to belong to each possible class [63]. They will be used in this thesis for different problem settings and therefore be introduced in more detail in the next chapter.

4 Prediction models

This chapter discusses the two main models used in this thesis: naïve Bayes and SVMs. The following notation will be used consistently throughout the chapter:

n is the number of training or test compounds,

$\mathbf{x}$ will be used to denote training or test compounds,

y denotes the target value, i.e., the class label or potency value, of a compound,

$\mathbf{x}^{(i)}, y^{(i)}$ is used to refer to the i'th compound and target value,

$\mathcal{Y}$ denotes the set of all possible labels,

D is the number of dimensions that represent one compound $\mathbf{x}$,

x_d refers to the d'th dimension of compound $\mathbf{x}$,

$\delta(a, b)$ will be the abbreviated notation for the function $\delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$

Unless stated otherwise, formulas containing $\mathbf{x}^{(i)}, y^{(i)}$ usually hold for all $i \in \{1, \ldots, n\}$; this information will be omitted for brevity.

4.1 Naïve Bayes

In LBVS, naïve Bayes classifiers are often used to predict biological activity. They are less frequently used for other prediction tasks, such as the prediction of physicochemical properties [52]. The naïve Bayes classifier is a generative model that uses Bayes' theorem to model the posterior probability $P(y|\mathbf{x})$:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \tag{1}$$

Here, $P(\mathbf{x}|y)$ is the class likelihood of compound $\mathbf{x}$ given class y, $P(y)$ is the prior probability of class y, and $P(\mathbf{x})$ is the evidence, i.e., the marginal probability for a certain

compound $\mathbf{x}$ [55]. Since the evidence of the same compound $\mathbf{x}$ in the denominator in equation (1) is constant, it is sufficient to estimate the prior and the class likelihood:

$$P(y|\mathbf{x}) \propto P(\mathbf{x}|y)P(y) \tag{2}$$

To classify new instances, they are assigned to the class with the maximum posterior probability:

$$y = \text{argmax}_{\hat{y}\in\mathcal{Y}} P(\hat{y}|\mathbf{x}) \tag{3}$$

The term *naïve* refers to the underlying assumption of descriptor independence, i.e., the class likelihood is modeled as a product of individual descriptor contributions [63]:

$$P(\mathbf{x}|y) = \prod_{d=1}^{D} P(x_d|y) \tag{4}$$

In practice, descriptor independence is usually not given. Therefore, it can make sense to perform a careful preprocessing of descriptors, e.g., via principal component analysis. However, it has also been shown that naïve Bayes can perform well also on correlated input data [64]. According to equation (3), the model parameters of naïve Bayes are the estimates of the class likelihood according to equation (4) and the prior. The prior can be either given if the probability distribution of the classes is known, or estimated from the training data as the fraction of samples from each class:

$$P(y) = \frac{\sum_{i=1}^{n} \delta(y^{(i)}, y)}{n} \tag{5}$$

However, the modeling of the individual descriptors' class likelihoods depends on the nature of the data [55]. If the descriptors are continuous and normally distributed, they are modeled using univariate Gaussians:

$$P(x_d = z|y) = \frac{1}{\sqrt{2\pi\sigma^2_{yx_d}}} \exp(-\frac{(z-\mu_{yx_d})^2}{2\sigma^2_{yx_d}}) \tag{6}$$

Hence, the mean μ_{yx_d} and variance $\sigma^2_{yx_d}$ of the descriptors x_d for each class y have to be computed from the training data using maximum likelihood estimation:

$$\mu_{yx_d} = \frac{1}{n_y} \sum_{i=1}^{n_y} \delta(y^{(i)}, y) x_d \tag{7}$$

$$\sigma^2_{yx_d} = \frac{1}{n_y} \sum_{i=1}^{n_y} \delta(y^{(i)}, y)(x_d - \mu_{yx_d})^2 \tag{8}$$

$$n_y = \sum_{i=1}^{n} \delta(y^{(i)}, y) \tag{9}$$

In the case of categorical descriptor values, the multinomial distribution is used:

$$P(x_d = z|y) = \prod_{i=1}^{n} p_{yz}^{\delta(\mathbf{x}^{(i)},z)} \tag{10}$$

In this case, p_{yz} is the joint probability of class y and descriptor value z, which is estimated as

$$p_{yz} = \frac{\sum_{i=1}^{n} \delta(y^{(i)}, y)\delta(x_d^{(i)}, z)}{\sum_{i=1}^{n} \delta(y^{(i)}, y)} \tag{11}$$

Finally, if all descriptors are binary, which is the case for molecular fingerprints, the Bernoulli distribution is used:

$$P(x_d = z|y) = \prod_{i=1}^{n} p_{yz}^{z}(1 - p_{yz})^{(1-z)} \tag{12}$$

Since in the binary case, $P(x_d = 0|y) = 1 - P(x_d = 1|y)$ holds, it is sufficient to estimate $P(x_d = 1|y)$:

$$P(x_d = 1|y) = \prod_{i=1}^{n} p_{yx_d} \tag{13}$$

$$= \frac{\sum_{i=1}^{n} \delta(y^{(i)}, y)\delta(x_d^{(i)}, x_d)}{\sum_{i=1}^{n} \delta(y^{(i)}, y)} \tag{14}$$

$$= \frac{\sum_{i=1}^{n} \delta(y^{(i)}, y)x_d^{(i)}}{\sum_{i=1}^{n} \delta(y^{(i)}, y)} \tag{15}$$

In practice, one usually applies Laplacian smoothing to equation (11) and equation (15) to prevent ill-defined probabilities for fingerprint bits that are always or never set. Then, the Laplacian smoothing factor α is the only hyperparameter that needs to be given; otherwise, naïve Bayes classification is hyperparameter-free.

In chapter 3 of this thesis, we will use naïve Bayes classification for the prediction of compound activity profiles. Here, the assumption of feature independence will be exploited to enable the training on incomplete data. Furthermore, chapter 5 introduces an interactive graphical representation for the interpretation of naïve Bayes classifiers using the Bernoulli distribution. For this purpose, the log odds ratio of $P(x_d = 1|y)$ is leveraged to explain both the complete model and individual classification decisions.

4.2 Support vector machines

Support vector machines (SVMs) are supervised, discriminative models that aim to separate instances from two classes [60]. As such, they are primarily designed for binary classification problems. However, formulations for regression and structured output

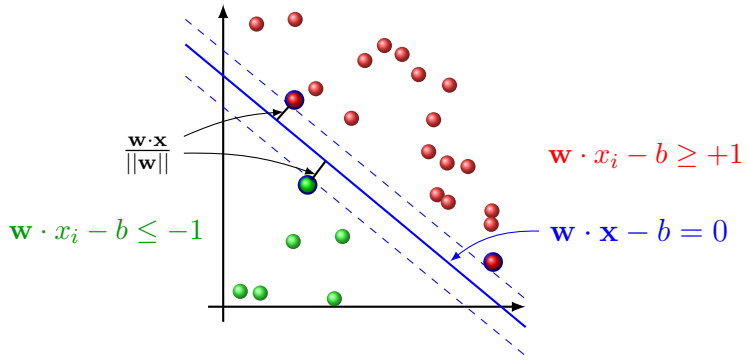


Figure 7: Schematic visualization of a linear SVM. The training examples of both classes are depicted using red and green circles, respectively. Support vectors are visualized using blue borders. The continuous and dashed blue lines represent the separating plane and its margins, respectively.

have also been proposed [65, 66]. Since SVMs are used for different types of problems throughout this thesis, all three SVM variants will be introduced in the following.

4.2.1 Classification

The concept of SVMs has originally been developed for binary classification of linearly separable data [60]. In the following years, extensions for inseparable training data, non-linear data, and imbalanced problems have been introduced [67–69]. Here, the linearly separable case is discussed first, and then the modifications for other use cases are briefly explained. A detailed derivation of the formulas used in the classification case can be found in the appendix.

Linearly separable data

The idea of an SVM is to separate two classes by a plane in high-dimensional space [60]. If the training labels y are expressed numerically in the set $\{-1, +1\}$, the plane should be able to separate all training instances such that the following holds for all training instances and labels:

$$y^{(i)}(\mathbf{w}\cdot\mathbf{x}^{(i)} - b) \geq 1 \tag{16}$$

Hence, the model parameters are the normal vector $\mathbf{w}$ and the bias b. New test instances are then classified by the side of the hyperplane they fall on, corresponding to the sign of the following function:

$$f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x} - b \tag{17}$$

If the data is separable according to equation (16), there are infinitely many hyperplanes that separate the data. Out of these, the optimal one is chosen, i.e., the one that

maximizes the distance between the closest training examples from different classes, the so-called *margin*. Figure 7 depicts a linearly separable 2D problem, where the margins are depicted by dashed lines. This leads to the primal optimization problem for linear maximum margin hyperplanes:

$$\min \frac{1}{2}\mathbf{w} \cdot \mathbf{w} \tag{18}$$

$$\text{subject to } y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) \geq 1 \tag{19}$$

This is a convex quadratic programming problem with only linear constraints and can as such be solved directly [70]. However, the elegance of SVMs lies in the expression of the problem in dual space. Without assuming convexity, the Lagrangian of equation (18) and equation (19) can be defined as [60, 71]:

$$\Lambda(\mathbf{w}, b, \lambda) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{n} \lambda^{(i)}[y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) - 1] \tag{20}$$

This function is maximized with respect to $\lambda^{(i)}$ with the additional constraints $\lambda^{(i)} \geq 0$ for all $\lambda^{(i)}$ [71]. Furthermore, it has to satisfy the Karush-Kuhn-Tucker (KKT) conditions [71]. If the KKT conditions and the partial derivatives of the Lagrangian are considered (see appendix for details), $\mathbf{w}$ can be expressed as:

$$\mathbf{w} = \sum_{\text{support vectors}} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} \tag{21}$$

Here, it is sufficient to consider those training examples where $\lambda^{(i)} > 0$, the so-called *support vectors*. This means that the number of summands in equation (21) can drop dramatically, which reduces both storage and computational requirements. The classification rule can then be expressed as the sign of:

$$f(\mathbf{x}) = \sum_{\text{support vectors}} \lambda^{(i)} y^{(i)} (\mathbf{x}^{(i)} \cdot \mathbf{x}) - b \tag{22}$$

The advantage of solving the dual instead of the primal optimization problem lies not only in the reduction of operations required for the final classification. It also enables two extensions that make SVMs especially powerful: the separation of (a) noisy and (b) nonlinear data.

Noisy data

In the case of noisy training data, it is not possible to separate all instances without error. Therefore, non-negative slack variables $\xi^{(i)}$ are introduced that allow some instances to be misclassified or lie inside the margin [67]. The primal optimization then

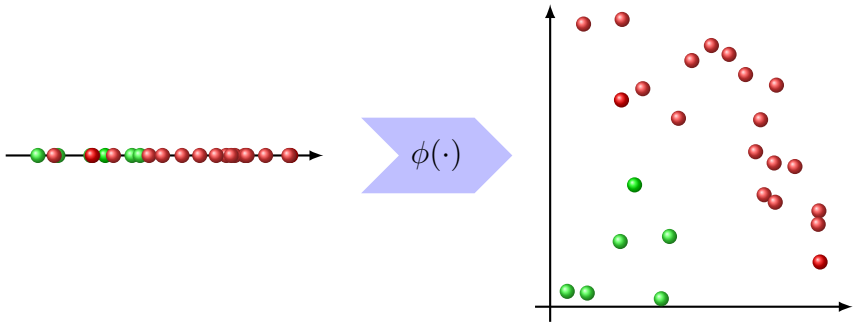


Figure 8: For problems that are not linearly separable, a mapping $\phi(\cdot)$ projects the data into a higher-dimensional space where linear separation becomes feasible.

changes to:

$$\min \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{n} \xi^{(i)} \tag{23}$$

$$\text{subject to } y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) \geq 1 - \xi^{(i)} \tag{24}$$

$$\xi^{(i)} \geq 0 \tag{25}$$

In this formulation, the regularization parameter C controls the trade-off between margin maximization and permitted amount of training error. As such, it has to be determined in advance and given to the algorithm as a hyperparameter.

If the dual problem is solved and the KKT conditions are considered, the slack variables and their corresponding dual variables ν vanish from the problem [67]. Altogether, it yields the same function as in the linearly separable case, which has to be maximized subject to:

$$\sum_{i=1}^{n} \lambda^{(i)} y^{(i)} = 0 \tag{26}$$

$$0 \leq \lambda^{(i)} \leq C \tag{27}$$

Hence, the computation of $\mathbf{w}$ and the classification rule stays the same as in the separable case (see appendix for details).

Nonlinear data

In the case of the data that is not linearly separable, the training instances $\mathbf{x}^{(i)}$ are projected into a higher-dimensional space by a mapping $\phi(\cdot)$ [68]. Then, $\mathbf{w}$ is no longer D-dimensional, but has the dimension of $\phi(\mathbf{x}^{(i)})$. Figure 8 exemplifies this idea using a mapping from 1D to 2D space. This change alters the Lagrangian as follows (see appendix for details):

$$\Lambda(\lambda) = \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)})) \tag{28}$$

Using Mercer's theorem [72], it is possible to provide a positive semidefinite *kernel function* $K(u, v)$ that implicitly calculates the inner product $\phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)})$. Then, the dual problem can be rewritten as:

$$\Lambda(\lambda) = \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \tag{29}$$

Hence, it is possible to derive and use the SVM model without explicitly computing the mapping $\phi(\cdot)$. However, there is one drawback: the normal vector $\mathbf{w}$ is expressed in the domain of $\phi(\cdot)$, which may be infinite. As a consequence, it cannot be computed explicitly anymore, making the interpretation of the resulting model hard or even impossible. Therefore, SVMs using kernels are often referred to as "black box" models [14].

Nevertheless, they are widely used in chemoinformatics for different problem settings [14]. Popular kernels include the linear, polynomial, sigmoid, and Gaussian or radial basis function (RBF) kernels:

$$K_{\text{linear}}(u, v) = u \cdot v \tag{30}$$

$$K_{\text{polynomial}}(u, v) = (a(u \cdot v) + b)^c \tag{31}$$

$$K_{\text{sigmoid}}(u, v) = \tanh(a(u \cdot v) + b) \tag{32}$$

$$K_{\text{Gaussian}}(u, v) = \exp(-\gamma ||u - v||^2) \tag{33}$$

Here, the parameters a, b, c, and γ are additional kernel parameters that have to be given as hyperparameters to the algorithm. In chemoinformatics, the Gaussian kernel is often chosen for nonlinear problems over the polynomial or sigmoid kernel [52]. Furthermore, a variety of kernel functions have been developed especially for the prediction of compound activity in LBVS [14]. One of the most widely applied kernels is the Tanimoto kernel, which was developed in accordance with the Tanimoto coefficient (Tc) [51, 73]:

$$K_{\text{Tanimoto}}(u, v) = \frac{u \cdot v}{u \cdot u + v \cdot v - u \cdot v} \tag{34}$$

The Tanimoto kernel is often used together with molecular fingerprints [52], because it is fast to compute on binary data and furthermore parameter-free. Other specialized kernel functions include pharmacophore kernels [74], target-ligand kernels [75], or structure-activity kernels [76].

Imbalanced Problems

In LBVS, often there are more inactive than active compounds available, inducing an imbalance of positive and negative training instances. For problem settings like this, Morik et al. [69] have suggested to use two regularization terms C_+ and C_- obeying the ratio:

$$\frac{C_+}{C_-} = \frac{|\{i | y^{(i)} = -1\}|}{|\{i | y^{(i)} = +1\}|} \tag{35}$$

C_+ and C_- are then used to balance the cost of slack variables associated with positive and negative training examples, respectively.

The minimization problem changes accordingly:

$$\min \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C_+ \sum_{\{i|y^{(i)}=+1\}} \xi^{(i)} + C_- \sum_{\{i|y^{(i)}=-1\}} \xi^{(i)} \tag{36}$$

Geometrically spoken, this alters the margin size: while it was symmetric in the balanced case, i.e., $\lambda^{(i)} \leq C$ for all i, the margin on the side of the minority class is now larger than the one on the majority classes side.

4.2.2 Regression

SVMs can also be used for regression, i.e., the prediction of real-valued target values [65]. In this case, a so-called ϵ-*insensitive loss function* is applied, which results in a loss of zero if the predicted value $f(\mathbf{x})$ deviates by less than ϵ from the expected target value y [60]:

$$|y - f(\mathbf{x})|_\epsilon = \begin{cases} 0 & \text{if } |y - f(\mathbf{x})| \leq \epsilon \\ |y - f(\mathbf{x})| - \epsilon & \text{otherwise} \end{cases} \tag{37}$$

For support vector regression (SVR), two sets of slack variables ξ and ξ_* are used to account for positive and negative deviations from the target values. This defines an "ϵ-tube" around the desired values in which misclassifications are not punished. Figure 9 visualizes this concept. The primal optimization problem for support vector regression (SVR) is given as [77]:

$$\min \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{n} (\xi^{(i)} + \xi_*^{(i)}) \tag{38}$$

$$\text{subject to } y^{(i)} - \mathbf{w} \cdot \mathbf{x}^{(i)} - b \leq \epsilon + \xi^{(i)} \tag{39}$$

$$\mathbf{w} \cdot \mathbf{x}^{(i)} + b - y^{(i)} \leq \epsilon + \xi_*^{(i)} \tag{40}$$

with nonnegative $\xi^{(i)}, \xi_*^{(i)}$. The regression function can then be written analogously to the classification case:

$$f(\mathbf{x}) = \sum_{\text{support vectors}} (\lambda^{(i)} + \lambda_*^{(i)}) K(\mathbf{x}^{(i)}, \mathbf{x}) + b \tag{41}$$

4.2.3 Structured output

The concept of SVMs has also been adjusted for the prediction of structured output [66, 78]. Here, the idea is to learn a function that maps the input vectors to complex output vectors. This is achieved via maximization over a discriminant function $F(\mathbf{x}, y, \mathbf{w})$:

$$f(\mathbf{x}, \mathbf{w}) = \arg\max_{y \in \mathcal{Y}} F(\mathbf{x}, y, \mathbf{w}) \tag{42}$$

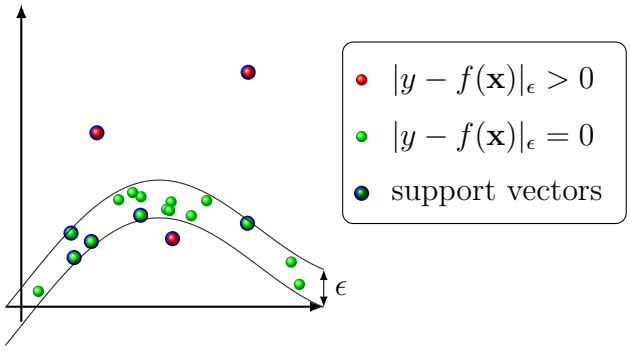


Figure 9: SVMs for regression fit an ϵ-insensitive tube through the data.

Here, $\mathbf{w}$ has the dimensionality of $\psi(\mathbf{x}, y)$, a combined feature representation of inputs and outputs that has to be defined specifically for the given problem. The optimization problem is given as [66]:

$$\min \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{n} \xi^{(i)} \tag{43}$$

$$\text{subject to } F(\mathbf{x}^{(i)}, y^{(i)}, \mathbf{w}) - F(\mathbf{x}^{(i)}, y, \mathbf{w}) \quad \geq 1 - \xi^{(i)} y \in \mathcal{Y} \setminus y^{(i)} \tag{44}$$

The constraints express that the discriminant function for the true output $y^{(i)}$ is at least $1 - \xi^{(i)}$ larger than for any other output. Furthermore, since the outputs y can be arbitrarily complex, a specialized loss function $\Delta(y, \hat{y})$ is required. Tsochantaridis et al. [66] propose two ways to incorporate this loss into the optimization: slack rescaling and margin rescaling. The constraints from equation (44) then change:

$$F(\mathbf{x}^{(i)}, y^{(i)}, \mathbf{w}) - F(\mathbf{x}^{(i)}, y, \mathbf{w}) \geq 1 - \frac{\xi^{(i)}}{\Delta(y^{(i)}, y)} \qquad \text{slack rescaling} \tag{45}$$

$$F(\mathbf{x}^{(i)}, y^{(i)}, \mathbf{w}) - F(\mathbf{x}^{(i)}, y, \mathbf{w}) \geq \Delta(y^{(i)}, y) - \xi^{(i)} \qquad \text{margin rescaling} \tag{46}$$

Again, this problem can be expressed in dual space, enabling the use of kernel functions. However, the number of constraints for structural SVMs is large with $n|\mathcal{Y}|$. In many cases, the output space $\mathcal{Y}$ can be very large, which in turn requires a larger number of training examples. Therefore, structural SVM problems are not always solvable by standard quadratic programming techniques. Tsochantaridis et al. [66] propose to use only a subset of constraints, which is chosen such that a "sufficiently accurate solution" is found [66]. In their algorithm, a working set of constraints is kept for every training example, and the dual problem is optimized using all constraints of these working sets. This process is iteratively repeated while constraints are added, until no further constraint is found which is violated more than some ϵ. The authors show that their algorithm finds a solution which is close to optimal [66], and provide an implementation in the publicly available SVM software SVM$^{\text{light}}$ [79]. In chapter 1, the structural SVM formalism is used to predict complete compound activity profiles, and compared to a set of individual classification SVMs.

4.3 Model interpretation

While many machine learning models have been shown to work well on a variety of problems related to drug discovery [14, 52], their interpretability strongly depends on the combination of molecular representation and learning algorithm. Models based on matched molecular pairs are often easily interpretable [80, 81], but their applicability is restricted to compounds forming MMP relationships. An example for a model based on MMPs will be given in chapter 2 of this thesis. While the resulting predictions are intuitively comprehensible, the discussed approach is only applicable to data sets of a certain constitution. On the other hand, models derived on the basis of molecular descriptors are applicable to any compound data set, but harder to interpret. Some machine learning algorithms, e.g., decision trees, can produce "rule sets" explaining the internal decision process of the model. However, these rules can become arbitrarily complex for large models. An advantage of molecular fingerprints is that it is possible to project each set bit back to the molecular graph [82–84]. This way, it is possible to visualize feature mappings in a way that is directly accessible for the medicinal chemist. In chapter 5 and chapter 6, we will use visual feature mappings to explain individual model decisions. However, these and similar methods require a measure of importance for each descriptor or fingerprint bit.

Whether individual feature contributions can be extracted for an individual model strongly depends on the learning algorithm. Individual decision trees and their ensemble in random forests for instance offer to assess the importance of each feature in terms of the number and order of splits they appear in. Feature contributions of naïve Bayes classification can be statistically measured by their log odds ratio, as will be done in chapter 5. For SVMs using the linear kernel, it is possible to compute the normal vector $\mathbf{w}$, which can be seen as a vector of weights for each dimension of the input. For ANNs with a single layer, a weight vector can also be computed. However, ANNs are most successful when they contain one or more hidden layers; the same holds for SVMs using kernels. These models can be extremely powerful, yet at the same time impossible to interpret in terms of the input representation. Still, interpretable models are of high interest, especially in life sciences where machine learning is often used to explain phenomena that are not completely theoretically understood.

One popular approach to the explanation of black box models is rule extraction by mimicry [85, 86]. Here, one first trains a successful, yet uninterpretable model like an ANN or SVM. In the next step, a highly intuitive learning algorithm is used with the aim not to model the original input data, but to mimic the complex model as closely as possible. The interpretable rules of this model are then thought to explain the workings of the black box predictor. In chapter 6, we will use a different approach to explain the classification decisions of SVMs. While our method is not as general as rule extraction approaches, it is able to directly disclose the inner workings of SVMs using the Tanimoto kernel on molecular fingerprints.

5 Thesis outline

This thesis is divided into three main parts. Part I describes the development of two methods for specialized use cases in LBVS. Herein, chapter 1 uses structural SVMs to model compound profiling experiments, and chapter 2 describes a new prediction method for hit expansion based on activity probabilities derived from matching molecular series. Next, part II reveals opportunities and challenges for machine learning applications in drug discovery. The first study in chapter 3 shows how the feature independence assumption of the naïve Bayes approach can be exploited to learn and predict on incomplete data. Furthermore, it is shown that the advent of publicly available chemogenomics data can be used for activity prediction, even in the absence of molecular structures. On the other hand, chapter 4 highlights limitations of SVR modeling for potency prediction. While these models may work well globally, they often fail to correctly predict the most potent, and therefore most important, compounds in the data sets. Finally, the topic of part III is the intuitive assessment and interpretation of LBVS models using molecular fingerprints. Here, we aim to bridge the gap between the highly active field of visual SAR visualization and the application of machine learning in drug discovery. Finally, conclusions are drawn and opportunities for future research are discussed.

Part I

Model Development for Pharmaceutical Tasks

1

Modeling of Compound Profiling Experiments Using Support Vector Machines

Introduction

The modeling of compound activity profiles is a complex task that becomes more and more important with the availability of chemogenomics data. In this study, we attempt to model compound profiling experiments using naïve Bayes classifiers and SVMs. For each compound, not only a single activity, but a range of activities against multiple targets is predicted. Since the number of possible compound activity profiles increases exponentially with the number of targets, this classification task is non-trivial. Furthermore, the public availability of complete compound profiling matrices is still limited, and activity profiling matrices are usually sparse in nature. Due to the complex character of the activity profiles, standard performance measures cannot be applied or have to be very carefully considered and analyzed.

To address these challenges, we develop and compare different classification models: a number of binary naïve Bayes and SVM models applied for each target individually, an SVR-based full profile classifier, and a profile predictor based on the structural SVM formalism. These models are applied to a set of 429 pyridinyl imidazole-based inhibitors that were screened against 24 different kinases.

Chem Biol Drug Des 2014; **84:** 75–85

Research Article

Modeling of Compound Profiling Experiments Using Support Vector Machines

Jenny Balfer[1,†], Kathrin Heikamp[1,†], Stefan Laufer[2] and Jürgen Bajorath[1,*]

[1]Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Dahlmannstr. 2, D-53113 Bonn, Germany
[2]Department of Pharmacy and Biochemistry, Pharmaceutical/Medicinal Chemistry, Eberhard-Karls-Universität Tübingen, Auf der Morgenstelle 8, D-72076 Tübingen, Germany
**Corresponding author: Jürgen Bajorath, bajorath@bit.uni-bonn.de*
[†]The contributions of these authors should be considered equal.

Profiling of compounds against target families has become an important approach in pharmaceutical research for the identification of hits and analysis of selectivity and promiscuity patterns. We report on modeling of profiling experiments involving 429 potential inhibitors and a panel of 24 different kinases using support vector machine (SVM) techniques and naïve Bayesian classification. The experimental matrix contained many different activity profiles. SVM predictions achieved overall high accuracy due to consistently low false-positive and consistently high true-negative rates. However, predictions for promiscuous inhibitors were affected by false-negative rates. Combined target-based SVM classifiers reached or exceeded the performance of SVM profile prediction methods and were superior to Bayesian classification. The classifiers displayed different prediction characteristics including diverse combinations of false-positive and true-negative rates. Predicted and experimentally observed compound activity profiles were compared in detail, revealing activity patterns modeled with different accuracy.

Key words: activity profile prediction, Bayesian classification, compound profiling, inhibitors, machine learning, protein kinases, support vector machines, target families

Received 2 December 2013, revised 6 January 2014 and accepted for publication 19 January 2014

Experimental testing of compound libraries against arrays of therapeutically relevant targets such as protein kinases or G protein-coupled receptors is often carried out in pharmaceutical research (1–4). Compound profiling against target families makes it possible to identify novel active compounds, assess their selectivity and promiscuity, and collect structure–activity relationship (SAR) information. Experimental evaluation of inhibitors across different kinase subfamilies has identified many promiscuous compounds (5) and a variety of activity and selectivity patterns (6,7). Profiling experiments are also of interest for computational analysis and design. For example, profiling data have been utilized to build models for the identification of kinase inhibitors (8,9) and promiscuous compounds (9). Furthermore, machine learning and similarity search methods have been applied to predict multiple activities of drugs and activity profiles (10–13).

In this study, we have modeled a kinase inhibitor profiling experiment and carried out systematic profile predictions using different support vector machines (SVMs) and Bayesian classification. The underlying complete experimental matrix contained 429 pyridinyl imidazole inhibitors assayed against 24 different kinase targets (14). In the following, the results of profiling matrix and individual activity profile predictions are reported.

Profiling Data Analysis

Data sets

For our analysis, a set of 429 compounds sharing a pyridinyl imidazole core (Figure 1) were used that were assayed against a panel of 24 different kinases (14).[a] These compounds represented potential ATP site-directed kinase inhibitors. A complete 429 × 24 activity matrix was obtained. By design, the library was focused on the p38-α kinase, but many imidazole derivatives displayed notable kinase differentiation potential, with small structural modifications leading to significant changes in activity profiles (14). Hence, the prediction of activity profiles comprising this profiling matrix was considered a challenging task.

Data preprocessing

Activities were measured as '% residual activity' at 10 μM compound concentration (14).[a] Hence, no IC_{50} or K_i was available, and it was expected that activity data were noisy. For computational modeling, '% residual activity' values were transformed into a binary activity readout (i.e.

 doi: 10.1111/cbdd.12294

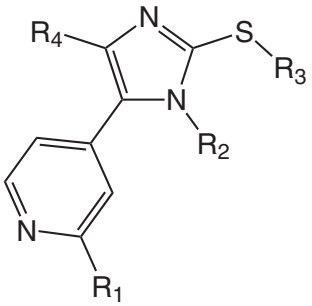


Figure 1: Inhibitor core structure. Shown is the pyridinyl imidazole structure upon which the compound library was based. R-group/substitution sites are indicated.

active versus inactive) applying a threshold value of a 20% residual activity (i.e. 80% inhibition) or, as a control, 40% residual activity (i.e. 60% inhibition). Compounds inhibiting a kinase below these thresholds were considered active (and inactive otherwise). The stringent 20% threshold was initially applied to exclude very weak kinase activities from consideration and put emphasis on activity profiles with strong inhibitory interactions. Figure 2 reports a small section of the binary data matrix. In this section, three compounds (mf286, mf265, mf258) only inhibit kinase p38-α, while three others (mf249, mf203, mf135) are active against several kinases. The remaining 9 compounds are inactive against all targets.

The binary matrix (20% threshold) yielded a total of 10 296 data points including 9384 inactive and 912 active ones. Hence, the activity matrix was overall sparsely populated with active data points. A total 115 of 429 compounds were inactive against all 24 kinases, and 106 compounds exclusively inhibited p38-α. Table 1 summarizes the most frequently observed activity profiles. There was a significant variety of profiles. In addition to the profiles reported in Table 1, 100 other profiles were found that were exhibited by only one or two compounds.

Molecular representations

For compounds, the MACCS[b] and ECFP4 (15) fingerprints were calculated (i.e. a fragment and a layered atom environment fingerprint) using the molecular operating environment (MOE).[c] Fingerprint overlap was quantified as a measure of molecular similarity by applying the Tanimoto coefficient (Tc) (16).

Although all compounds shared the same (small) structural core (Figure 1), their calculated similarities yielded a wide range of Tc values, as illustrated in Figure 3. More than half of the compounds yielded Tc values of < 0.5 using MACCS, which corresponds to a low range (17). Similar observations were made for ECFP4 where Tc values of < 0.2 were frequently observed.

Taken together, the following data characteristics made this profiling matrix a challenging test case for machine learning: There was significant variation in activity profiles and small structural changes often led to significant activity profile variations.

Support Vector Machine Theory

SVM modeling

In previous studies, SVM models were derived to distinguish between compounds with related activities (18) and closely related ligands with different mechanisms of action (19). In our current analysis, we applied different SVM variants including a 'structural SVM' that has formerly been

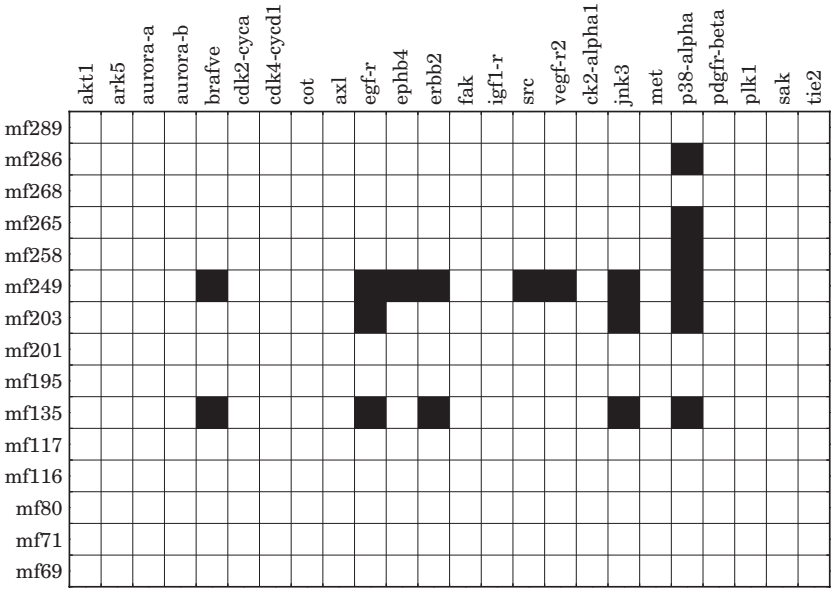


Figure 2: Binary data matrix. A small section of the binary data matrix comprising 15 compounds is shown. White squares indicate inactivity, black squares activity.

Table 1: Most frequently occurring activity profiles

Inhibited kinases	No. ligands
None	115
p38-alpha	106
brafve, p38-alpha	31
igf1-r, src, p38-alpha	11
igf1-r, src	8
egf-r	8
jnk3, p38-alpha	8
egf-r, p38-alpha	7
egf-r, igf1-r, src, vegf-r2	4
brafve, egf-r, erbb2, p38-alpha	4
brafve, egf-r, p38-alpha, tie2	4
brafve, src, p38-alpha	4
brafve	4
egf-r, erbb2, p38-alpha	3
p38-alpha, tie2	3
egf-r, erbb2, jnk3, p38-alpha	3
egf-r, erbb2, src, jnk3, p38-alpha, sak, tie2	3
brafve, egf-r, erbb2, src, vegf-r2, jnk3, p38-alpha, tie2	3

The most frequent kinase activity profiles of data set compounds are reported for the stringent 80% inhibition threshold.

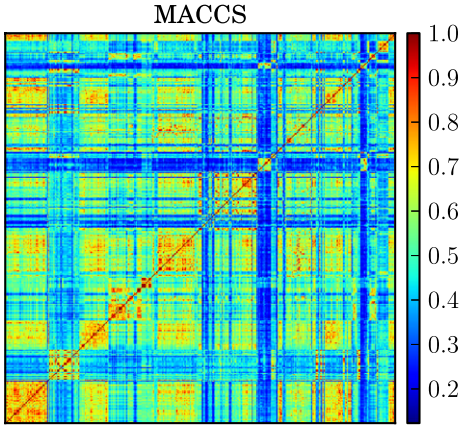


Figure 3: Pairwise Tanimoto similarities. The matrix reports MACCS Tanimoto coefficient values for pairwise comparison of all library compounds.

used for biological sequence comparison (20–22) and compound ranking against single targets (23,24).

In the following, a brief account of SVM theory relevant for our study is provided. SVMs are discriminative models that attempt to derive a hyperplane $(w \cdot x) - b = 0$ to best separate objects with different class labels (e.g. 'positive' or 'negative') (25). Therefore, SVMs must be trained using n training instances with known labels.

Classification SVM

The classification SVM uses the hyperplane $(w \cdot x) - b = 0$ to separate positive from negative instances (25). A new instance x_i can be classified depending on the side of the hyperplane it falls: x_i is considered positive if $(w \cdot x_i) - b > 0$ and negative otherwise. Assuming the binary labels -1 for negative and $+1$ for positive training instances, the optimal hyperplane can be constructed by minimizing

$$\phi(w) = \frac{1}{2}(w \cdot w)$$

subject to

$$y_i[(w \cdot x_i) - b] \geq 1, i = 1, 2, \ldots, n$$

It is possible that the training data are not separable without errors. Therefore, so-called *slack* variables $\xi_i \geq 0$ are introduced (26), and a soft-margin separating hyperplane is derived by minimizing

$$\phi(w, \xi) = \frac{1}{2}(w \cdot w) + C\sum_{i=1}^{n} \xi_i$$

subject to

$$y_i[(w \cdot x_i) - b] \geq 1 - \xi_i, i = 1, 2, \ldots, n$$

In this equation, C is a given parameter that penalizes large slack variables; if it is small, large ξ_i are permitted and many training errors are tolerated. If C is large, small ξ_i are favored and training errors are suppressed by learning a more complex hyperplane. To enable arbitrary complex decision boundaries, the inner product $(w \cdot x)$ is replaced by a kernel function $K(u,v)$ (27). Kernel functions implicitly generate the inner product of u and v in a high-dimensional space, thereby circumventing the need to explicitly map u and v to this space.

Regression SVM

It is also possible to utilize SVMs for the estimation of regression functions, that is, the prediction of real values. Instead of predicting a positive class if $(w \cdot x) - b > 0$, the value of $(w \cdot x) - b$ is predicted. To determine w and b, the empirical risk $R(w, b)$ is minimized over the n training examples (25):

$$R(w,b) = \frac{1}{n}\sum_{i=1}^{n} |y_i - (w \cdot x_i - b)|_\varepsilon$$

Here, the subscript ε indicates that $R(w,b)$ describes the ε-insensitive loss function, that is, the loss is considered zero if $|y_i - (w \cdot x_i - b)| < \varepsilon$ (25). Similar to the classification SVM, the optimization problem is solved by minimizing

$$\phi(w, \xi^*, \xi) = \frac{1}{2}(w \cdot w) + C\left(\sum_{i=1}^{n} \xi_i^* + \sum_{i=1}^{n} \xi_i\right)$$

subject to

$$\begin{matrix} y_i - (w \cdot x_i) - b \leq \varepsilon + \xi_i^* \\ (w \cdot x_i) + b - y_i \leq \varepsilon + \xi_i \end{matrix}, i = 1, 2, \ldots, n$$

For SVM regression, two sets of non-negative slack variables are required to account for both the positive and the negative deviation of the output value from its expected value.

Structural SVM

Other non-numeric outputs can also be predicted. In contrast to classification and regression, structural SVMs (21) learn a function f depending on w and x that is maximal for the best output y:

$$f(x,w) = \arg\max_{y \in Y}(w \cdot \Psi(x,y))$$

Here, $\Psi(x,y)$ is a combined feature representation of inputs and outputs. The underlying idea of structural SVMs is that for any input and output, a function can be learned that will yield a maximal value for the correct input/output pair and smaller values for all incorrect pairings. Therefore, structural SVMs do not necessarily separate positive and negative pairs by a maximum margin hyperplane, but rather learn a scoring scheme for input/output pairs. The structure of the combined feature representation $\Psi(x,y)$ is generally unknown; it is problem-specific and has to be specifically implemented for each individual application.
The minimization problem is then given as

$$\min \frac{1}{2}(w \cdot w) + C\sum_{i=1}^{n}\xi_i$$

subject to

$$\forall y \in Y \backslash y_i : w \cdot (\Psi(x_i,y_i) - \Psi(x_i,y)) \geq 1 - \xi_i, i = 1,2,\ldots,n$$

As the structure of Y is unknown and the space of all possible $y \in Y$ can be infinitive, there is no trivial solution to this minimization problem. In fact, the minimization problem contains one constraint for each $y \in Y$ and can practically only be solved for a restricted subspace of Y. Therefore, Tsochantaridis *et al.* (21) introduced an algorithm that only considers a polynomially-sized subset of constraints and thereby makes the solution of the minimization feasible. Thus, a function is required to determine the most violated of all constraints:

$$\hat{y} = \arg\max_{y \in Y} H(y)$$

where $H(y)$ represents the cost of predicting y if the true label is y_i. If it is possible to provide a combined feature representation $\Psi(x, y)$ of inputs and outputs and, in addition, determine the most violated constraint for a given problem, structural SVMs can be generated.

Profiling Matrix Prediction

For modeling profiling experiments, several SVM modeling strategies can be applied. First, binary SVM classifiers can be derived for each target, as illustrated in Figure 4A.

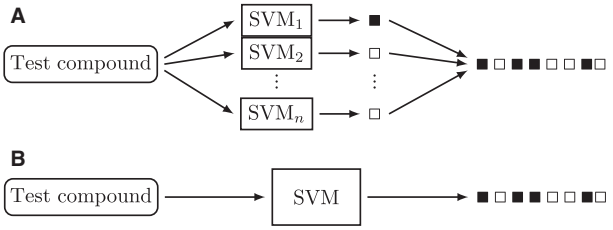


Figure 4: SVM classification principles. (A) The target-based classifier uses 24 binary SVMs to predict the activity of a compound against each individual target. The resulting predictions are then combined to yield the activity profile. (B) Both the full profile and the svmstruct classifier use a single SVM model to directly predict the activity profile of a compound.

While this target-based (or per target) classifier is an elegant way to simplify the complex problem of predicting complete activity profiles, it does not take correlations between different kinase activities into account. Second, SVM models can be derived to directly predict activity profiles for given compounds, as illustrated in Figure 4B. One approach is based on SVM regression and termed full profile classifier, the other is based on the structural SVM and termed svmstruct classifier.

Target-based classifier

Twenty-four different SVM classification models were trained to predict the activity of each compound–target combination. Individual predictions were then combined to generate the complete activity profile of each compound. For training and prediction, the Tanimoto kernel (28) was used to compare two compound fingerprints u and v:

$$K_{\text{Tanimoto}}(u,v) = \frac{u \cdot v}{u \cdot u + v \cdot v - u \cdot v}$$

Full profile classifier

The full profile prediction derived only a single SVM model to predict the complete profiling matrix. During training, many different activity profiles were generated for each compound and used in combination with its fingerprint as inputs x_i. The outputs y_i were the regression scores obtained by the sum of correct profile positions (bits) minus the sum of incorrect bits in the generated profile. Thus, the regression value was maximal for the true profile of a training compound and minimal for the inverted true profile. For prediction, a number of possible profiles were generated, and the one yielding the highest regression score was selected. Because the input data x_i contained the compound's structure and its activity profile, we used a combined structure-profile kernel (13,29) for SVM training and prediction:

$$K_{\text{combined}}(u,v) = K_{\text{Tanimoto}}(u_{\text{compound}}, v_{\text{compound}}) \times (\phi(u_{\text{profile}}) \cdot \phi(v_{\text{profile}}))$$

where $\phi(x)$ is a function that maps a binary activity profile into a higher-dimensional space representing the correlation of profile positions:

$$\phi(x) = \begin{pmatrix} [t_0 = -1] & [t_0 = -1] \\ [t_0 = -1] & [t_0 = +1] \\ [t_0 = -1] & [t_1 = -1] \\ \vdots & \vdots \\ [t_n = +1] & [t_n = +1] \end{pmatrix}$$

Here, $[t_i = a][t_j = b]$ is 1 if position i has value a and position j value b for $a,b \in \{-1, +1\}$; otherwise, the value is 0 (30).

Svmstruct model

The svmstruct approach yields one model to predict a profiling matrix, which is distinct from the full profile model. As combined input representation $\Psi(x,y)$, compound structure and activity profile were required and hence the combined structure-profile kernel was used. To identify the most violated constraint for svmstruct, a greedy approximation was applied that produced different profiles and selected the constraint associated with the profile yielding the highest cost $H(y)$.

Naïve Bayesian classification

We have compared the SVM predictions to naïve Bayesian classifiers (NB) (31), a popular activity prediction approach (32,33). These classifiers are based on Bayes theorem and have often been successfully applied in the context of in silico target and activity prediction (33).

Calculation Setup

Calculation protocol

To predict the complete profiling matrix, we carried out 10 independent trials for each combination of an SVM strategy, fingerprint and inhibition threshold. For each trial, 100 training compounds were randomly selected and used to generate an SVM model. The model was then used to predict the activity profiles of the remaining 329 compounds. If the same position in the profiling matrix was predicted differently in independent trials, the final prediction was determined via a consensus voting, that is, the most frequent prediction was considered the final one. In addition, positions with the same number of positive and negative predictions were classified as inconclusive. The libraries *SVM*^{light} and *SVM*^{struct} (34) were used to build the different models. Standard parameters were applied, with one exception: The regularization parameter $C = 1000$ was consistently used to generate SVMs (34). For NB control calculations, the freely available Python package *scikit-learn* (35) was used. The Bernoulli naïve Bayes formulation was applied to account for the binary nature of fingerprint representations. NB calculations were carried out using different prior probabilities. First, uniform prior probabilities were applied, that is the same prior probabilities were used for positive and negative predictions. Second, prior probabilities were determined from training data, that is, these prior probabilities were set to account for the ratio of active and inactive data set compounds.

Performance evaluation

For each individual compound–kinase interaction, it was determined whether a prediction was a true negative (TN, predicted and experimentally inactive), true positive (TP, predicted and experimentally active), false negative (FN, predicted inactive but experimentally active), or false positive (FP, predicted active but experimentally inactive). On the basis of this categorization, overall 'balanced' and 'unbalanced' prediction accuracy was determined as:

$$A_{\text{unbalanced}} = \frac{\#\text{TN} + \#\text{TP}}{\#\text{actives} + \#\text{inactives}}$$

$$A_{\text{balanced}} = \frac{0.5 * \#\text{TP}}{\#\text{actives}} + \frac{0.5 * \#\text{TN}}{\#\text{inactives}}$$

The accuracy A is the fraction of individual matrix entries that were correctly predicted and thus provides a measure of the overall quality of the predictions. In contrast to unbalanced prediction accuracy, balanced accuracy equally weights TPs and TNs, regardless of the composition

Table 2: Global prediction statistics

	Experimental data	
	Active	Inactive
Target-based		
Active	405	164
Inactive	461	9154
Inconclusive	46	66
Full profile		
Active	256	60
Inactive	629	9300
Inconclusive	27	24
SVM structure		
Active	340	132
Inactive	532	9204
Inconclusive	40	48
Naïve Bayes		
Active	370	328
Inactive	514	8970
Inconclusive	28	86

Reported is the absolute number of true positives, true negatives, false positives, and false negatives for different prediction methods. For example, 'active/experimentally active' reports true positives and 'active/experimentally inactive' false positives. Activity profile positions with the same number of positive and negative predictions over all 10 trials were classified as 'inconclusive'.

of data sets and distribution of actives versus inactives. However, globally determined *A* cannot account for the quality of individual compound profile predictions. Simple numerical values are unable to account for different combinations of TNs, TPs, FNs, and FPs in predicted activity profiles. Therefore, predicted activity profiles were analyzed in detail.

Results and Discussion

Descriptor evaluation

The full profiling matrix had 429 × 24 = 10 296 entries, which were systematically predicted using all classifiers with two fingerprints as descriptors. The performance of the models was very similar for the alternative descriptors. The only notable difference was that the use of ECFP4 resulted in an overall larger number of FN predictions (likely due to its smaller pairwise compound similarity values compared to MACCS). Therefore, we report the results obtained with MACCS.

Global prediction accuracy

The first round of predictions was carried out applying a threshold value of 80% inhibition to the compound data (see Methods). The target-based classifier correctly predicted 9559 of 10 296 entries, yielding an unbalanced accuracy of 92.84% and a balanced accuracy of

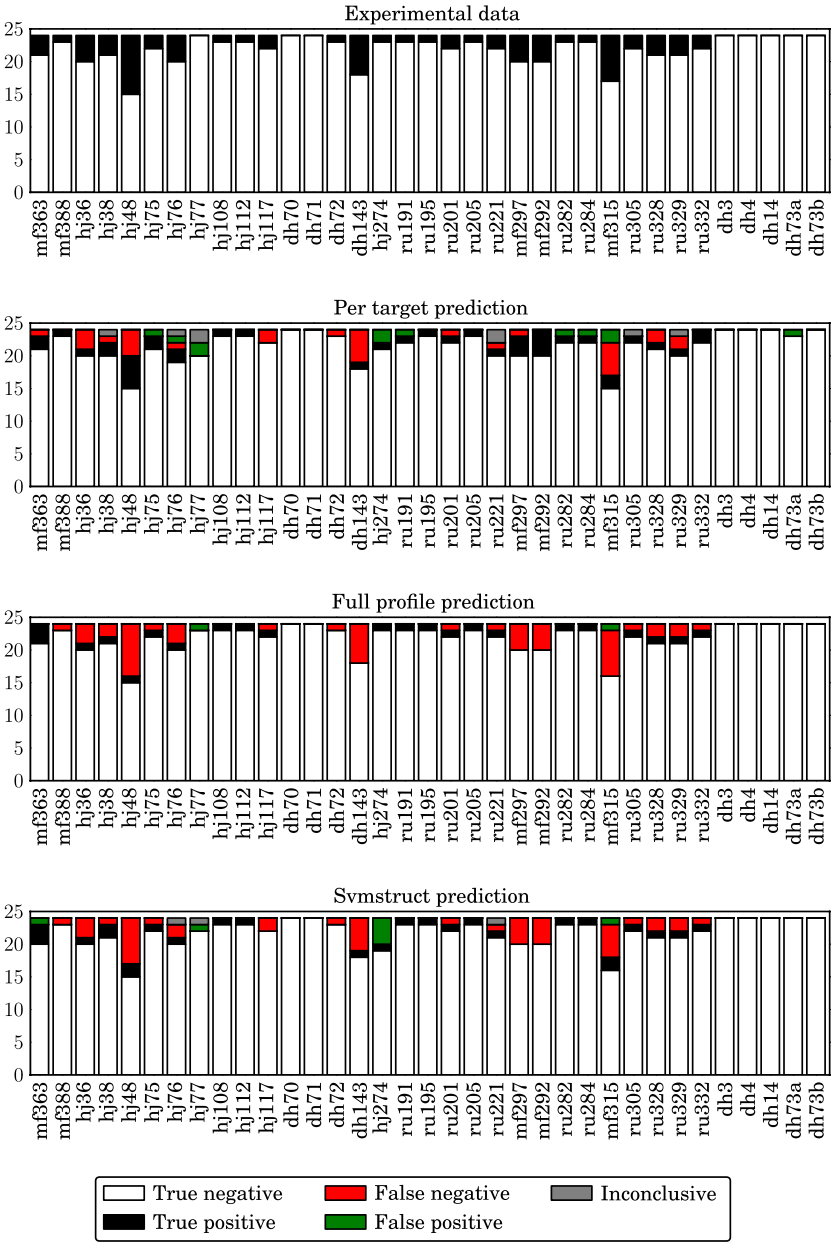


Figure 5: Prediction accuracy. Bars report the total number of true and false positives, true and false negatives, and the number of inconclusive bits in predicted compound activity profiles.

70.98%. The full profile classifier produced 9556 correct predictions (92.81% unbalanced accuracy versus 63.59% balanced accuracy) and the svmstruct classifier 9544 (92.70% unbalanced, 67.68% balanced accuracy). The naïve Bayesian classifier reached a global accuracy of 90.71% (unbalanced) and 68.08% (balanced) using uniform prior probabilities, and of 92.15% (unbalanced) and 64.96% (balanced) using prior probabilities determined from data. Generally high unbalanced prediction accuracy was achieved by producing only very few FN predictions. When the threshold value was lowered to 60% (resulting in more active compounds), unbalanced prediction accuracy was reduced by 6.84% (target-based SVM classifier) to 8.2% (NB with data-dependent priors). The balanced accuracy was improved by 2.61% (target-based SVM classifier) to 7.96% (full profile classifier).

Because the NB classifier using uniform prior probabilities for active and inactive predictions resulted in a slightly higher balanced accuracy, these values are discussed in the following.

Table 2 reports absolute numbers of TNs, TPs, FNs, and FPs for the different classifiers. Despite comparable accuracy, different calculation characteristics emerged. The target-based SVM classifier predicted TPs at a high rate, with 405 correctly predicted inhibitors, compared to only 256 for the full profile and 340 for the svmstruct classifier. NB also correctly predicted 370 inhibitory interactions, but nearly doubled the FP rate of the target-based SVM classifier (with 328 versus 164 FPs). Moreover, the more complex SVM models even further reduced false-positive rates, with 132 and only 60 FPs for svmstruct and the full profile classifier, respectively. Overall, the SVM and NB target classifier predicted most TPs, but also yielded most inconclusive predictions, whereas the more complex SVM classifiers had slightly higher TN rates and lower FP rates. TN rates were lowest and FP rates highest for NB classification.

Activity profile analysis

Global accuracy assessment did not provide information regarding the activity profile locations and contexts of incorrect predictions. Did misclassifications predominantly affect a subset of compounds or were limited prediction errors widely distributed over compounds? How did predicted profiles compare in detail to experimentally observed ones? To answer these questions, activity profiles were analyzed in detail.

Figure 5 reports true and false positives and negatives for a representative subset of 35 compounds. The chart at the top shows the number of targets the compounds were active or inactive against. Many ligands were consistently inactive or only inhibited a small number of kinases. In Figure 5, exceptions included compounds hj48, dh143 and mf315, which were active against nine, six, and seven kinases, respectively. The other three charts report the number of TNs, TPs, FNs, FPs, and inconclusive positions in the predicted activity profiles. The majority of predicted interactions were TNs. However, FNs also consistently occurred throughout the profiles of all SVM classifiers. There were often more FNs than TPs, which points at a general weakness of the predictions. By contrast, FPs and inconclusive predictions were rare for all classifiers.

Figure 6 shows experimental profiles for a subset of 12 ligands and their SVM predictions. Profiles with low activity density were mostly correctly predicted. For instance,

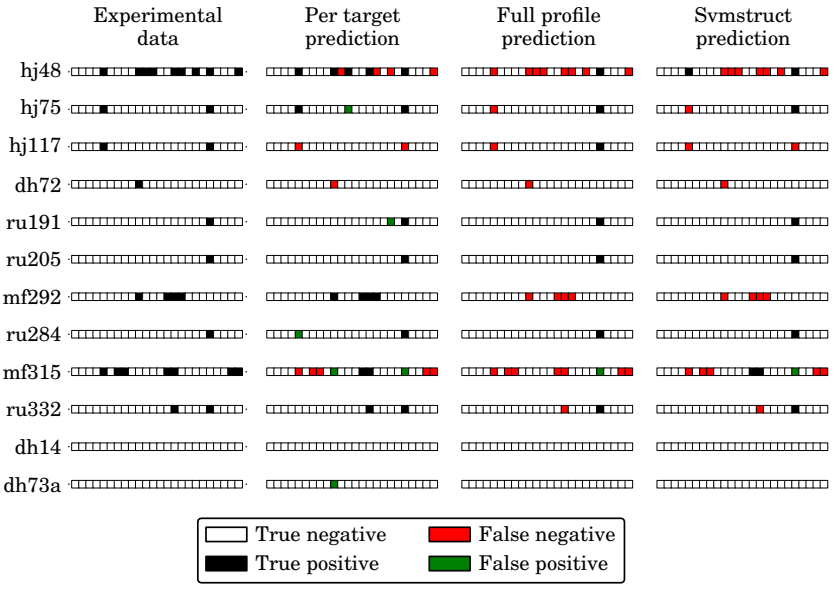


Figure 6: Exemplary observed and predicted activity profiles. Experimental profiles of inhibitors with no, single, or multiple kinase activity are shown and compared to profiles predicted using different SVM models. The target-based classifier yields the lowest number of false-negative predictions but slightly more false-positive predictions than profile SVM models.

profiles of compounds ru191, ru205, ru284, dh14, and dh73a were correctly reproduced by all SVM classifiers. Three predictions using target-based classifier (ru191, ru284, and dh73a) yielded a single FP, consistent with the observation that target-based classification had a higher FP rate than the SVM profile models. Most predicted profiles did not contain FPs. FNs were detected in profiles of compounds with multikinase activities such as hj48 or mf315. Learning such multikinase profiles was difficult for SVMs because such profiles were underrepresented in the data set. In the case of compound hj48, predictions using the target-based, full profile, and svmstruct classifier yielded four, eight, and seven FNs, respectively. By contrast, the profile of mf292 (another compound with multikinase activity) was correctly predicted by the target-based classifier and incorrectly by the others.

Profile prediction characteristics

Figures 5 and 6 provide representative views of the profile predictions. Compound profiles with only few activity annotations were generally correctly predicted, whereas predictions of profiles with multiple kinase activities were more challenging and prone to FNs. Nonetheless, all classifiers were able to predict a variety of profiles. The target-based SVM classifier accurately predicted more different profiles than the other classifiers.

Figure 7 reports all combinations of true- and false-negative and positive predictions for the classifiers when 80% (Figure 7A) and 60% (Figure 7B) inhibition thresholds were applied. Each circle marks an observed combination of active/inactive profile positions and is scaled in size by the number of profiles having this combination. In the experimental matrix (top graph), most profiles

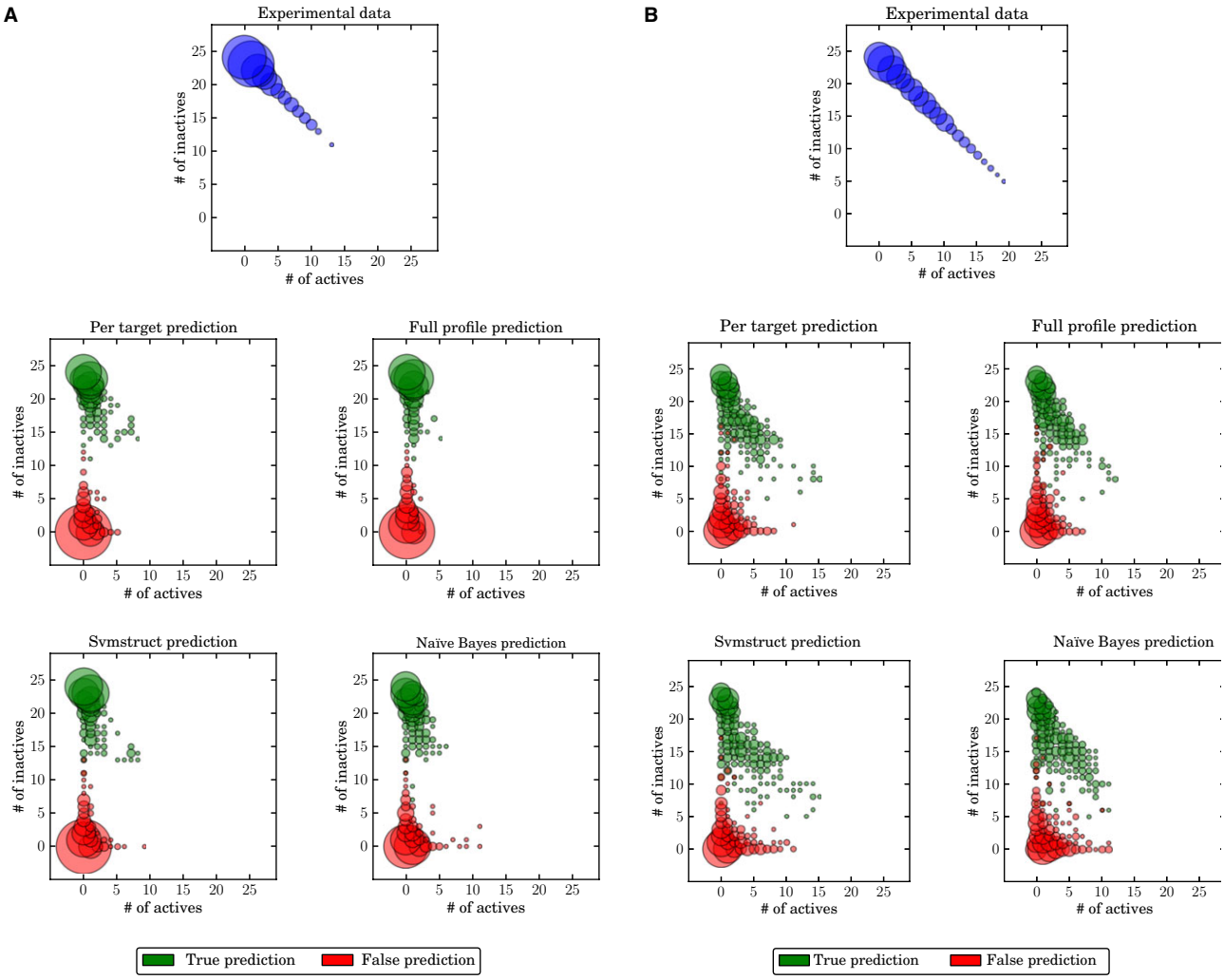


Figure 7: Profile predictions. Graphs reporting the count of active versus inactive profile positions are shown for data sets resulting from the application of a threshold of (A) 80% or (B) 60% inhibition. The size of the circles scales with the number of profiles having a given composition. In the upper graph, the distribution of experimental profiles (blue) is reported. The other four graphs report true (green) and false (red) predictions for the different SVM and NB models.

contained at most one positive positions and hence 24 or 23 negative positions. Most predicted profiles contained no or very few FPs and also only small number of FNs. Figure 7A shows that profiles with more than nine positive positions were not reproduced in any calculation when the stringent inhibition threshold was applied and also reveals further differences between the classifiers. NB calculations produced only a low number of profiles without any false predictions and predicted profiles with more than 10 FPs, which was not observed for SVMs. The full profile SVM classifier produced fewer FPs than the others, but also fewer TPs. In addition, the svmstruct classifier correctly predicted a slightly larger number of profiles with more than five active bits than the target-based classifier. Applying the less stringent 60% inhibition threshold significantly reduced the number of profiles with no kinase activity and resulted in profiles with more than 15 active positions. (Figure 7B). In addition, a larger number of activity profiles occurred only once, making predictions more challenging. All classifiers predicted profiles with larger numbers of active positions, thereby increasing TP and also FP rates (Figure 7B). All methods failed to correctly predict profiles with more than 16 active positions. As observed for the more stringent inhibition threshold, the NB classifier produced the lowest number of profiles without any prediction errors and the SVM full profile classifier yielded fewer FPs and TPs than the other classifiers. Hence, for both inhibition thresholds, equivalent profile prediction characteristics were observed. For the lower inhibition threshold, more active positions were consistently predicted. In both cases, SVM performance was superior to NB classification and the target-based SVM classifier exceeded the performance of profile prediction methods.

Conclusions

Compound profiling data are rarely publicly available, and there are not many opportunities for the evaluation of machine learning methods to predict activity profiles and profiling matrices. We have modeled a recently reported kinase profiling experiment using different SVM methods including target- and regression-based as well as structural SVM classifiers. In addition, target-based NB classification, a popular approach in target prediction, was carried out as a reference.

The experimental profiling matrix provided a challenging task for machine learning and predictions. Activity profiles significantly varied, but kinase activities were only sparsely distributed over the matrix. Small structural differences between data set compounds that shared a pyridinyl imidazole core often led to significant activity profile changes. In addition, only single-concentration percent inhibition activity data were available. Therefore, we applied different inhibition thresholds to convert these data into a binary format for modeling.

Activity profile predictions revealed systematic trends. SVM classifiers reached more than 90% global accuracy because FP and TN rates were consistently low and high, respectively. A general weakness of the models was that less than half of the available TPs were detected. When a low inhibition threshold was applied, TP and FP rates increased. Importantly, target-based SVM classifier reached or exceeded the performance of the profile classifiers and yielded higher prediction accuracy than NB calculations. The profile prediction characteristics of all models varied, as revealed by detailed activity profile analysis. SVM profile classifiers also accurately predicted many profiles containing only small numbers of active positions and had very low FP rates.

The results also have implications for practical applications. Serially applied target SVM classifiers can be effectively utilized to prescreen candidate compounds for activity profiling including libraries. On the basis of our findings, these classifiers would have a high potential to effectively eliminate candidates having a low propensity to be active against individual kinases. This would be especially relevant when searching for specific kinase inhibitors when compound promiscuity should be low and activity annotations sparsely distributed over profiling matrices. Models reported herein should thus be useful when the pyridinyl imidazole or related chemical libraries are further expanded in the search for specific kinase inhibitors.

Acknowledgments

The authors thank Dilyana Dimova and Norbert Furtmann for discussions and help with the data set.

Conflict of Interest

The authors state no conflict of interest and have received no payment for preparation of this manuscript.

References

1. Fabian M.A., Biggs W.H. III, Treiber D.K., Atteridge C.E., Azimioara M.D., Benedetti M.G., Carter T.A. *et al.* (2005) A small molecule-kinase interaction map for clinical kinase inhibitors. Nat Biotechnol;23:329–336.
2. Allen J.A., Roth B.L. (2011) Strategies to discover unexpected targets for drugs active at G protein-coupled receptors. Annu Rev Pharmacol Toxicol;51:117–144.
3. Kenakin T. (2008) Functional selectivity in GPCR modulator screening. Comb Chem High Throughput Screen;11:337–343.

4. Bi K., Lebakken C.S., Vogel K.W. (2011) Transformation of in vitro tools for kinase profiling: keeping an eye over the off-target liabilities. Expert Opin Drug Discov;6:701–712.
5. Goldstein D.M., Gray N.S., Zarrinkar P.P. (2008) High-throughput kinase profiling as a platform for drug discovery. Nat Rev Drug Discov;6:391–397.
6. Metz J.T., Johnson E.F., Soni N.B., Merta P.J., Kifle L., Hajduk P.J. (2011) Navigating the kinome. Nat Chem Biol;7:200–202.
7. Milletti F., Hermann J.C. (2012) Targeted kinase selectivity from kinase profiling data. ACS Med Chem Lett;3:383–386.
8. Martin E., Mukherjee P. (2012) Kinase-kernel models: accurate in silico screening of 4 million compounds across the entire human kinome. J Chem Inf Model;52:156–170.
9. Niijima S., Shiraishi A., Okuno Y. (2012) Dissecting kinase profile data to predict activity and understand cross-reactivity of kinase inhibitors. J Chem Inf Model;52:901–912.
10. Kawai K., Fujishima S., Takahashi Y. (2008) Predictive activity profiling of drugs by topological-fragment-spectra-based support vector machines. J Chem Inf Model;48:1152–1160.
11. Peragovics Á., Simon Z., Brandhuber I., Jelinek B., Hári P., Hetényi C., Czobor P., Málnási-Csizmadia A. (2012) Contribution of 2D and 3D structural features of drug molecules in the prediction of drug profile matching. J Chem Inf Model;52:1733–1744.
12. Simon Z., Peragovics Á., Vigh-Smeller M., Csukly G., Tombor L., Yang Z., Zahoránszky-Kőhalmi G., Végner L., Jelinek B., Hári P., Hetényi C., Bitter I., Czobor P., Málnási-Csizmadia A. (2012) Drug effect prediction by polypharmacology-based interaction profiling. J Chem Inf Model;52:134–145.
13. Jacob L., Vert J.-P. (2008) Protein-ligand interaction prediction: an improved chemogenomics approach. Bioinformatics;24:2149–2156.
14. Dimova D., Iyer P., Vogt M., Totzke F., Kubbutat M.H.G., Schächtele C., Laufer S., Bajorath J. (2012) Assessing the target differentiation potential of imidazole-based protein kinase inhibitors. J Med Chem;55:11067–11071.
15. Rogers D., Hahn M. (2010) Extended-connectivity fingerprints. J Chem Inf Model;50:742–754.
16. Rogers D.J., Tanimoto T.T. (1960) A computer program for classifying plants. Science;132:1115–1118.
17. Vogt M., Bajorath J. (2011) Introduction of the conditional correlated bernoulli model of similarity value distributions and its application to the prospective prediction of fingerprint search performance. J Chem Inf Model;51:2496–2506.
18. Heikamp K., Bajorath J. (2013) Prediction of compounds with closely related activity profiles using weighted support vector machine linear combinations. J Chem Inf Model;53:791–801.
19. Balfer J., Vogt M., Bajorath J. (2013) Searching for closely related ligands with different mechanisms of action using machine learning and mapping algorithms. J Chem Inf Model;59:2252–2274.
20. Joachims T., Galor T., Elber R. (2006) Learning to align sequences: a maximum-margin approach. In: Leimkuhler B., Chipot C., Elber R., Laaksonen A., Mark A., Schlick T., Schütte C., Skeel R., editors. New Algorithms for Macromolecular Simulation, Vol. 49. Berlin Heidelberg: Springer; p. 57–69.
21. Tsochantaridis I., Hofmann T., Joachims T., Altun Y. (2004) Support vector machine learning for interdependent and structured output spaces. In: Proc of the 21st International Conference on Machine Learning, p. 104–111.
22. Yu C.-N.J., Joachims T., Elber R., Pillardy J. (2007) Support vector training of protein alignment models. In: Proc of the 11th Annual International Conference on Research in Computational Mol Biol, p. 253–267.
23. Rathke F., Hansen K., Brefeld U., Müller K.-R. (2010) StructRank: a new approach for ligand-based virtual screening. J Chem Inf Model;51:83–92.
24. Varnek A., Baskin I. (2012) Machine learning methods for property prediction in chemoinformatics: quo vadis? J Chem Inf Model;52:1413–1437.
25. Vapnik V.N. (1995) The Nature of Statistical Learning Theory. New York: Springer.
26. Cortes C., Vapnik V.N. (1995) Support-vector networks. Mach Learn;20:273–297.
27. Boser B.E., Guyon I.M., Vapnik V.N. (1992) A training algorithm for optimal margin classifiers. In: Proc of the 5th Annual Workshop on Computational Learning Theory.
28. Ralaivola L., Swamidass S.J., Saigo H., Baldi P. (2005) Graph kernels for chemical informatics. Neural Netw;18:1093–1110.
29. Heikamp K., Hu X., Yan A., Bajorath J. (2012) Prediction of activity cliffs using support vector machines. J Chem Inf Model;52:2354–2365.
30. Joachims T., Finley T., Yu C.-N.J. (2009) Cutting-plane training of structural SVMs. Mach Learn;77:27–59.
31. Duda R.O., Hart P.E., Stork D.G. (2000) Pattern Classification, 2nd edn. New York: Wiley-Interscience; 20–83 p.
32. Bender A., Mussa H.Y., Glen R.C., Reiling S. (2004) Molecular similarity searching using atom environments, information-based feature selection, and a naïve Bayesian classifier. J Chem Inf Comput Sci;44:170–178.
33. Koutsoukas A., Lowe R., KalantarMotamedi Y., Mussa H.Y., Klaffke W., Mitchell J.B.O., Glen R.C., Bender A. (2013) In silico target predictions: defining a benchmarking data set and comparison of performance of the multiclass naïve Bayes and Parzen-Rosenblatt window. J Chem Inf Model;53:1957–1966.

34. Joachims T. (1999) Making large-scale support vector machine learning practical. In: Schölkopf B., Burges C.J.C., Smola A.J., editors. Advances in Kernel Methods. Cambridge, MA: MIT Press; p. 169–184.
35. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M. *et al.* (2011) Scikit-learn: machine learning in python. J Mach Learn Res;12:2825–2830.

Notes

[a]ProQinase Free Choice Biochemical Kinase Assays. http://www.proqinase.com/ (accessed October 15, 2013).
[b]MACCS Structural Keys, Accelrys: San Diego, CA, 2011.
[c]Molecular Operating Environment (MOE), version 2012.10, Chemical Computing Group, Montreal, QC, 2012.

Summary

In this study, different machine learning models were built and developed for the task of compound activity profiling. Profiling compounds against a library of related targets is often carried out in the lead optimization stage of the drug discovery process. This way, it is possible to assess the activity probability of a compound target interaction, if it is known that this compound is active or inactive against a related target. Furthermore, compound profiling can be carried out for the analysis of compound promiscuity and selectivity.

Our results show that the models produced quite different profile predictions. Most inactive compound target combinations were correctly reproduced, while the underrepresented class of active compound target interactions was underpredicted by all classifiers. Furthermore, the combination of individual target-based binary SVMs achieved the highest prediction accuracy.

The following chapter deals with the development of a probabilistic prediction method for hit expansion. In contrast to finding similar compounds that are active against related targets, here we seek to discover compounds with increasing structural diversity that are all active against the same target. Furthermore, the predictions should be comprehensible and chemically intuitive to enable prediction-driven compound design.

2

Hit Expansion from Screening Data Based upon Conditional Probabilities of Activity Derived from SAR Matrices

Introduction

The goal of the previous study was to identify compounds with characteristic promiscuity or selectivity patterns against related targets. In this chapter, the development of a probabilistic prediction method for hit expansion is described. Here, one or several promising hits have already been identified. The goal is to expand the chemical space around them, i.e., to find compounds with increasing structural diversity in the chemical neighborhood of existing hits. These neighbors should be identified by defined chemical transformations to enable the synthesis by substitution of certain R-groups. Furthermore, it is desirable for the predictions to be intuitively explainable, enabling medicinal chemists to make informed decisions about the compounds to synthesize next.

To address these needs, we develop a conditional probability-based prediction method, where the probabilities are derived from SAR matrices. SAR matrices are data structures that organize MMPs, i.e., compounds with related core and substructure fragments. Hence, predictions derived by the described method are chemically interpretable and the prioritized compounds have defined transformation pathways from existing ones.

The data selection, preprocessing, and study design was carried out by Disha Gupta-Ostermann. My main contribution is the formal derivation of the conditional probability framework, which is described in the following section. The full study has been published as follows:

> Gupta-Ostermann, D.; Balfer, J.; Bajorath, J. Hit Expansion from Screening Data Based upon Conditional Probabilities of Activity Derived from SAR Matrices. *Mol. Inf.* **2015**, *34*, 134–146.

Fundamentals

An SAR matrix is a data structure that organizes compounds from matching molecular series (MMS) in a 2D table [87]. First, all MMPs with a single exchanged substructure are derived from a set of compounds, and these are then subjected to another round of fragmentation. This approach gives rise to several sets of related MMS, each of which consists of several compounds with a shared core. All series whose cores again form an MMS are then organized together in an SAR matrix. Figure 10 shows an exemplary SAR matrix derived from three MMS. Here, each filled cell represents one compound from the data set, colored by its activity annotation. Since not all exchanged substructures are present in every single MMS in the matrix, some cells are empty. These empty cells represent so-called *virtual compounds*, which have not yet been synthesized. Their activity is to be predicted using the information from the SAR matrix.

A given compound data set can contain an arbitrary number of related MMS, leading to a possibly large number of SAR matrices. Furthermore, there can be matrices where the number of virtual compounds largely exceeds the number of actual data set compounds. Other matrices consist exclusively of active or inactive compounds. Both sparsely populated matrices and matrices with a single class label are not suitable for our modeling approach. In the full study, we have therefore filtered the matrices that are considered [88].

Contribution

The SAR matrix-based prediction approach derives activity probabilities for each core and substructure occurring in an MMS. To predict the activity probability of a virtual compound, the probabilities of its core and substructure are combined. The underlying idea is that the higher a core's or substructure's activity probability, the more it influences the prediction of a virtual compound. To give an example, if all data set compounds with a certain core are active, then a virtual compound with the same core will most likely also be active. On the contrary, if half of data set compounds with a certain core are active and half of them are inactive, the exchanged substructures of the virtual compound must strongly influence its binding affinity. This concept is encoded in a probabilistic framework, which is schematically shown in figure 11.

As a first step, core and substructure class probabilities are derived by estimating the conditional probabilities $P(y|\mathbf{c})$ and $P(y|\mathbf{s})$ from the training data. Here, $\mathbf{c}^{(i)}$ and $\mathbf{s}^{(i)}$ are used to refer to the core and substructure of compound i, respectively.

$$P(y|\mathbf{c}) = \frac{\sum_{i=1}^{n} \delta(\mathbf{c}^{(i)}, \mathbf{c})\delta(y^{(i)}, y)}{\sum_{i=1}^{n} \delta(\mathbf{c}^{(i)}, \mathbf{c})} \tag{47}$$

$$P(y|\mathbf{s}) = \frac{\sum_{i=1}^{n} \delta(\mathbf{s}^{(i)}, \mathbf{s})\delta(y^{(i)}, y)}{\sum_{i=1}^{n} \delta(\mathbf{s}^{(i)}, \mathbf{s})} \tag{48}$$

Figure 11 (a) reports the core and substructure class probabilities of the matrix from figure 10. While the substructure in the second column has a probability of 1 for the

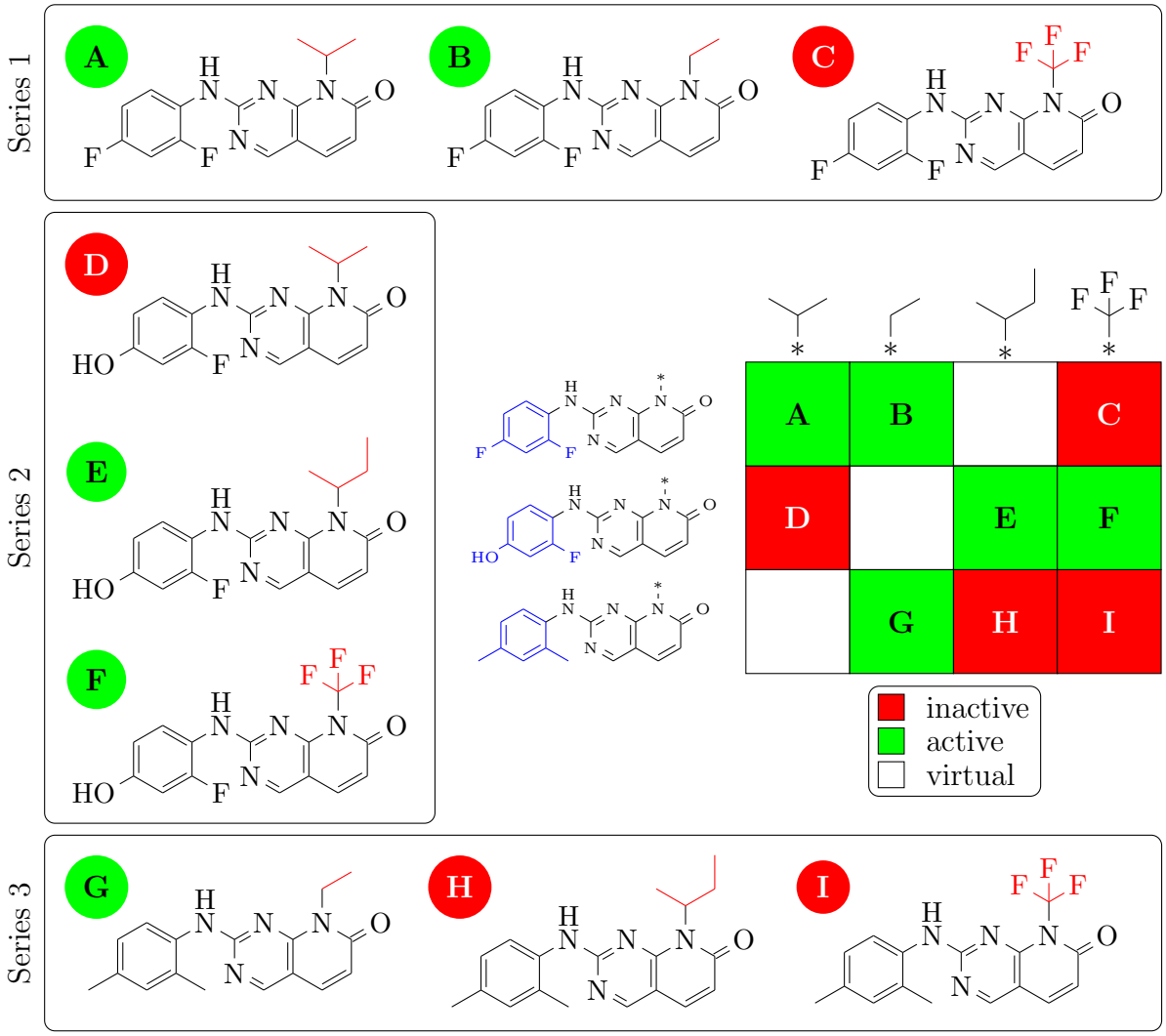


Figure 10: An exemplary SAR matrix derived from nine compounds forming three MMS. The rows of the matrix correspond to the common cores of each series, and each column represents one exchanged substructure. Exchanged substructures in each series and cores in the matrix are highlighted in red and blue, respectively. The figure is adapted from the original publication [88].

active and 0 for the inactive class, both probabilities are uniform for the third column's substructure. Hence, it is likely that the virtual compound in the second column will be active, whereas no information whatsoever can be inferred about the virtual compound of the third column if only the exchanged substructure is considered.

The major step of the proposed method is therefore to derive core and substructure class contributions, which transfer the uncertainty of the matrices' cores onto the substructures and vice versa. To recall the underlying concept, a virtual compound's core should have a high contribution to the prediction if its substructure cannot determine activity. On the other hand, if it is clear that a core determines the activity of a virtual compound, its substructure does not have to be considered. In the example, the virtual compound in the second column is likely to be active just because it contains the second substructure, whereas the activity of the virtual compound in the third column has to be determined by its core. This idea is formalized by first deriving class-dependent core and a substructure weights, respectively, and use these to compute class contributions. The core and substructure weights, $w(\mathbf{c}, y)$ and $w(\mathbf{s}, y)$, are simply given by their inverse class probabilities. The core weights are then used to calculate the substructure class contributions and vice versa:

$$w(\mathbf{c}, y) = P(y|\mathbf{c})^{-1} \tag{49}$$

$$w(\mathbf{s}, y) = P(y|\mathbf{s})^{-1} \tag{50}$$

$$c(\mathbf{c}, y) = \frac{\sum_{i=1}^{n} w(\mathbf{s}, y)\delta(\mathbf{c}^{(i)}, \mathbf{c})\delta(y^{(i)}, y) + \alpha}{\delta(\mathbf{c}^{(i)}, \mathbf{c}) + 2\alpha} \tag{51}$$

$$c(\mathbf{s}, y) = \frac{\sum_{i=1}^{n} w(\mathbf{c}, y)\delta(\mathbf{s}^{(i)}, \mathbf{s})\delta(y^{(i)}, y) + \alpha}{\delta(\mathbf{s}^{(i)}, \mathbf{s}) + 2\alpha} \tag{52}$$

Here, Laplacian smoothing is applied using a hyperparameter α. This prevents ill-defined probabilities for the cases when a certain core or substructure never appears in training compounds of a certain class.

In equation (51) and equation (52), the calculation of the core class contribution $c(\mathbf{c}, y)$ using the substructure weight $w(\mathbf{s}, y)$ and vice versa is the key aspect. It formulates the principle that a core contributes more to the final prediction of one class if the weights of the corresponding substructures are high, i.e., the class is underrepresented in the substructures. An example is the core class contribution of the second core and the active class in figure 11 (c). First of all, the majority of training compounds with the second core are active. Second, and most importantly, the substructures of the active training compounds with the second core do not contribute to an active prediction. While the third substructure does not convey any information about activity (i.e., $P(y = \text{active}|\mathbf{s}^{(3)}) = P(y = \text{inactive}|\mathbf{s}^{(3)}) = 0.5$), the presence of the fourth substructure is an indicator for inactivity ($P(y = \text{active}|\mathbf{s}^{(4)}) < P(y = \text{inactive}|\mathbf{s}^{(4)})$). Therefore, the contribution of core $\mathbf{c}^{(2)}$ is likely to render the respective training compounds active, which is reflected in a high core class contribution $c(\mathbf{c}^{(2)}, \text{active})$.

Finally, the core and substructure class contributions are normalized to arrive at a final estimate of core and value class probabilities, denoted by $P'(y|\mathbf{c})$ and $P'(y|\mathbf{s})$, respectively. The final class probability of a virtual compound, $P(y|\mathbf{x}^{(i)})$, is then predicted

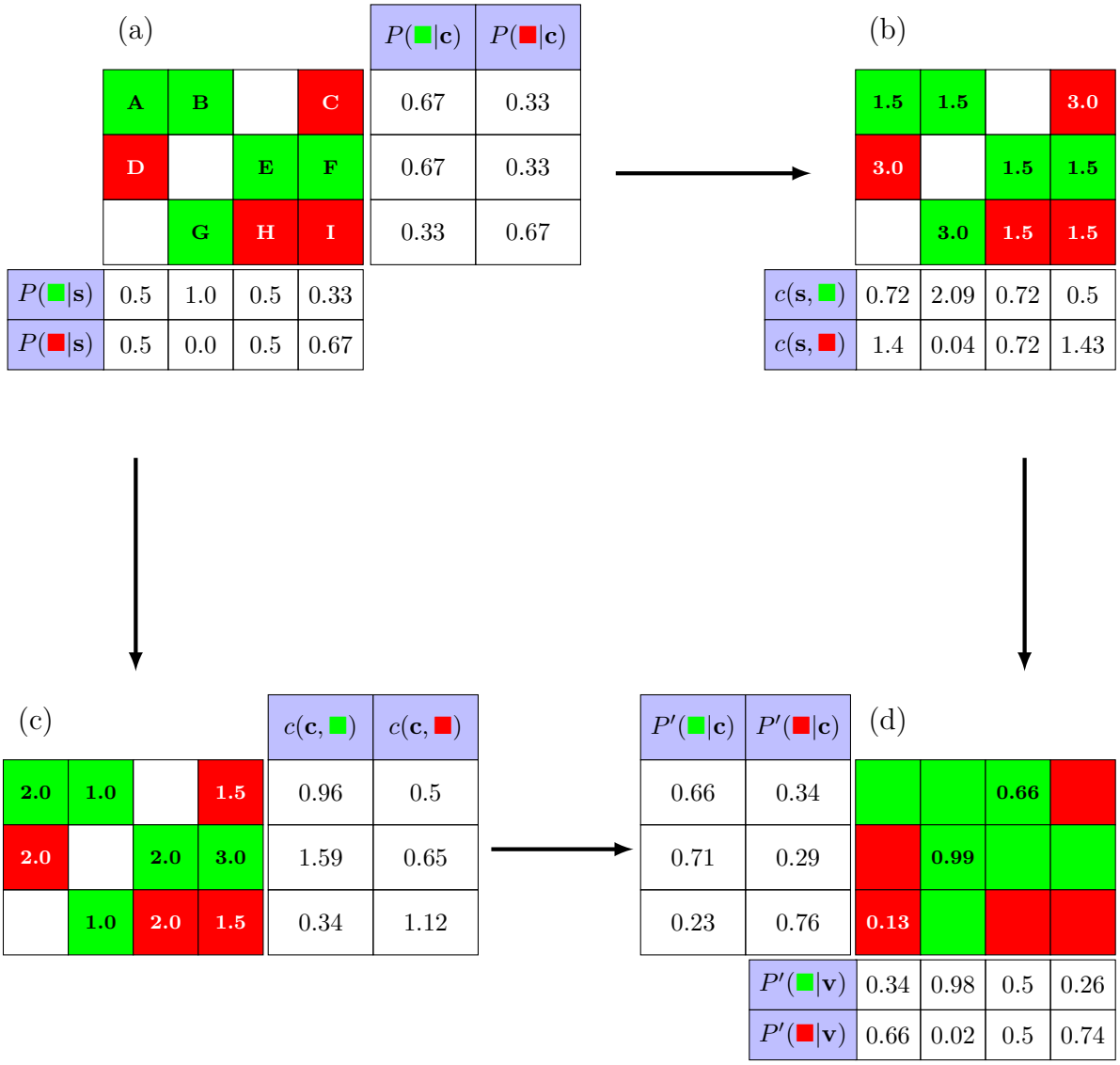


Figure 11: Schematic illustration of the SAR matrix-based prediction approach. Shown are (a) the derivation of core and substructure class probabilities, (b) the class-dependent core weights (in the matrix cells) and the substructure contributions derived from them, (c) the class-dependent substructure weights (in the matrix cells) and the core contributions derived from them, (d) the final estimates of core and value class probabilities and the activity probabilities for the virtual compounds. The figure is adapted from the original publication [88].

as a combination of both:

$$P'(y|\mathbf{c}) = \frac{c(\mathbf{c}, y)}{\sum_{\hat{y}\in\mathcal{Y}} c(\mathbf{c}, \hat{y})} \tag{53}$$

$$P'(y|\mathbf{s}) = \frac{c(\mathbf{s}, y)}{\sum_{\hat{y}\in\mathcal{Y}} c(\mathbf{s}, \hat{y})} \tag{54}$$

$$P(y|\mathbf{x}^{(i)}) = \frac{P'(y|\mathbf{c}^{(i)})P'(y|\mathbf{s}^{(i)})}{\sum_{\hat{y}\in\mathcal{Y}} P'(\hat{y}|\mathbf{c}^{(i)})P'(\hat{y}|\mathbf{s}^{(i)})} \tag{55}$$

Exemplary class probabilities for the active class are shown in figure 11 (d).

We have compared this approach to classifications derived by naïve Bayes classification, random forests, and SVMs using ECFP4 fingerprints. In many cases, comparable performances could be achieved [88]. Furthermore, the SAR matrix-based approach is easily interpretable by visualization of the matrices that contributed to a prediction.

Summary

This chapter derived a formalism to extract conditional activity probabilities from SAR matrices, and explained how they can be utilized to predict activities of untested compounds. In the original study, we have organized nine different compound data sets in SAR matrices, and carried out systematic benchmark calculations using our approach. Furthermore, the same compound data sets were represented using ECFP4 fingerprints, and predictions were derived using naïve Bayes classifiers, random forests, and SVMs. Our results show that the probabilistic SAR matrix-based approach performs comparable to these state-of-the-art machine learning algorithms [88]. These methods are however not generally comparable, because the matrix approach requires specific structural relationship between data set compounds. Furthermore, the results of the matrix-based predictions are intuitively explainable as chemical transformations and their contributions.

This and the last chapter have focused on the application and development of supervised learning algorithms for specific use cases in drug discovery. Beyond these, the next part of this thesis contains two studies that provide novel insights into machine learning when applied to pharmaceutically relevant questions. The focus is not on the development of new methods, but on the understanding of machine learning algorithms and their workings in the drug discovery context.

Part II

Insights into Machine Learning in Chemoinformatics

3

Compound Structure-Independent Activity Prediction in High-Dimensional Target Space

Introduction

In the last part, two methods for the prediction of compound activity profiles and single-target compound activities, respectively, were introduced. The focus of these studies was on method development and benchmarking of these methods. The following part, however, emphasizes insights into computational methods for compound activity prediction that can be gained by careful analysis. In this chapter, we introduce a method for compound structure-independent activity prediction in the presence of high-dimensional profiling data. As such, our study highlights a "paradigm-shift" in chemoinformatics that becomes feasible in the presence of publicly available chemogenomics data: compound activity is not only predicted based on the structures (traditional SAR), but also based on their activities against other targets [89, 90].

This chapter compares naïve Bayes classifiers based on compound structures with those based on compound activity profiles. The naïve Bayes approach is chosen because it can be adjusted to treat incomplete target annotations, which usually limits activity-based predictions in practical scenarios. The results are compared to structure-based SVMs and a naïve Bayes hybrid approach designed to incorporate both structure and activity information. All classifiers are applied to a high-dimensional profiling data set with compound annotations against 383 kinases. Interestingly, the activity-based naïve Bayes classifier is able to outperform the other methods in the presence of this high-dimensional target space. An in-depth feature analysis furthermore reveals the influence of different kinase activity annotations on the prediction accuracy of other kinases.

Full Paper

www.molinf.com

molecular informatics

DOI: 10.1002/minf.201400051

Compound Structure-Independent Activity Prediction in High-Dimensional Target Space

Jenny Balfer,[a] Ye Hu,[a] and Jürgen Bajorath*[a]

Abstract: Profiling of compound libraries against arrays of targets has become an important approach in pharmaceutical research. The prediction of multi-target compound activities also represents an attractive task for machine learning with potential for drug discovery applications. Herein, we have explored activity prediction in high-dimensional target space. Different types of models were derived to predict multi-target activities. The models included naïve Bayesian (NB) and support vector machine (SVM) classifiers based upon compound structure information and NB models derived on the basis of activity profiles, without considering compound structure. Because the latter approach can be applied to incomplete training data and principally depends on the feature independence assumption, SVM modeling was not applicable in this case. Furthermore, iterative hybrid NB models making use of both activity profiles and compound structure information were built. In high-dimensional target space, NB models utilizing activity profile data were found to yield more accurate activity predictions than structure-based NB and SVM models or hybrid models. An in-depth analysis of activity profile-based models revealed the presence of correlation effects across different targets and rationalized prediction accuracy. Taken together, the results indicate that activity profile information can be effectively used to predict the activity of test compounds against novel targets.

Keywords: Activity prediction · Multi-target activities · Pharmaceutical research · Naïve Bayesian models

1 Introduction

Assessing multi-target activities of compounds through target profiling has become an important exercise in pharmaceutical research.[1–5] In a typical profiling experiment, a collection of compounds is screened against an array of related or distinct targets. Compound profiling may have varying goals including, for example, the identification of active compounds that are selective for a target of interest over others (e.g., closely related targets and/or anti-targets) or the assessment of compound promiscuity across a given target family. In addition, compound profiling might also be carried out to establish compound-based relationships between multiple targets, which is of particular interest for chemical biology or chemogenomics.[6]

Regardless of specific goals, compound profiling often focuses on important therapeutic targets. In addition to G protein coupled receptors,[1] protein kinases are one of the major target families subjected to profiling.[2–5] Given the high interest in protein kinases as drug targets, many inhibitors directed against the ATP (cofactor) binding site in kinases have been developed over the past decade, especially for use in cancer treatment.[2,7] Because the ATP binding site in kinases is largely considered, many ATP site-directed inhibitors are active against multiple kinases,[2,7] although such inhibitors might also display a considerable degree of selectivity for individual kinases or subfamilies.[8] Profiling of compounds against kinases, especially of ATP site-directed inhibitors, is often carried out to better understand their promiscuity or selectivity patterns across the kinome and select inhibitors with higher or lower degrees of promiscuity for specific therapeutic applications.[7]

In the pharmaceutical industry, such profiling experiments typically produce large volumes of data, which are only rarely released into the public domain,[4] given their mostly confidential nature. In academia, compound profiling is less common (often due to resource constraints, rather than lack of interest). Consequently, there is only limited availability of compound profiling data in the public domain. At the same time, there is high interest in the chemoinformatics community in the development and assessment of computational approaches to model profiling experiments and predict multi-target compound activities, for several reasons. First, from a methodological perspective, multi-target activity prediction is a topic of interest for advanced machine learning approaches.[9] In addition, further extension of existing experimental data through interaction

[a] *J. Balfer, Y. Hu, J. Bajorath*
Department of Life Science Informatics, Bonn-Aachen International Center for Information Technology, Rheinische Friedrich-Wilhelms-Universität Bonn
Dahlmannstr. 2, D-53113 Bonn,Germany
tel: +49-228-2699-306; fax: +49-228-2699-341
**e-mail: bajorath@bit.uni-bonn.de*

Supporting Information for this article is available on the WWW under http://dx.doi.org/10.1002/minf.201400051.

predictions is also of considerable interest, especially for chemogenomics.[7,8,10] Furthermore, methods for activity profile prediction can be practically applied to aid in the study of drug polypharmacology[11] including the prediction of alternative drug targets,[12] side effects,[13] or new therapeutic applications.[14] Given the attractiveness of these applications, it is not surprising that a number of computational investigations have attempted multi-target activity predictions.[15–20]

From a methodological point of view, experimental profiling data can also serve as an input to compound activity prediction. Instead of predicting compound-target interactions based on chemical structures, which represents a conventional approach in chemoinformatics, bioactivity-based descriptors of compounds have been increasingly used in the past years.[21–25] For example, the use of bioactivity fingerprints, also called "affinity fingerprints"[21,25] or "high-throughput fingerprints",[23,24] for activity prediction has been explored, often leading to promising results in virtual screening benchmark calculations.[23,25] The derivation of such bioactivity-based descriptors typically requires large amounts of activity measurements for test compounds on multiple targets and has thus predominantly been explored in the pharmaceutical industry.

Despite limited public availability of profiling data, first efforts have also been made to analyze and predict kinase profiling campaigns.[19,20] In a recent study, it was attempted to reproduce a kinase profiling experiment involving a set of 429 pyridinyl-imidazole ATP site-directed inhibitors tested against a panel of 24 different kinases.[20] Because the design of these kinase inhibitors was primarily focused on the p38-α (MAPK14) kinase, a popular cancer target, the compounds were often inactive against distantly related kinases. Hence, active data points were sparsely distributed over the profiling matrix.[20] Nonetheless, a significant variety of compound profiles was observed. Under these conditions, serially applied target-based support vector machine (SVM) classifiers using structural fingerprint descriptors for compounds predicted profiles more accurately than profile-based regression or structural SVM models and naïve Bayesian (NB) classifiers.[20]

In this study, we have addressed a principally different activity prediction task using a kinase profiling data set consisting of 72 reference inhibitors including marketed drugs and 383 kinases covering major parts of the kinome. In this case, profiling data were available for only a limited number of compounds but many kinases, thus representing a high-dimensional target space with only little compound coverage. In our analysis, we have found that NB models derived by learning from activity profile data yielded more accurate activity predictions than compound structure-based SVM or NB classifiers, although profiling data available for learning was incomplete. Prediction accuracy based on activity profiles could not be further improved (or was reduced) using hybrid models that combined activity profile and compound structure information. Taken together, the results indicate that profile information is sufficient for activity predictions in high-dimensional target space, i.e., the activity of compounds against novel targets could be accurately predicted on the basis of activity profiles for other targets.

2 Materials and Methods

2.1 Kinase Profiling Data Set

We manually assembled a high-dimensional profiling data set with the aid of DiscoveRx's[26] KINOMEscan data and kinase interaction maps,[27] which are based upon a study by Davis et al.[28] The interaction maps report profiling data for 72 known kinase inhibitors including marketed drugs from a large-scale kinase screen.[27] According to the experimental protocol,[26] all compounds were first tested at a single concentration screen of 10 μM. Subsequently, K_d values were determined for all compounds that yielded less than 35% of positive control activity for a given kinase target. K_d values obtained for pairwise compound-kinase interactions ranged from 20 pM to 2.9 μM.

To assemble a profiling data set, we retrieved the structures of tested inhibitors from the ChEMBL database,[29] standardized the structures, and downloaded their available activity annotations from TREEspot kinase interaction maps.[26,27] These efforts resulted in a complete profiling matrix for the 72 inhibitors and a total of 383 different kinase targets belonging to 11 subfamilies, hence providing extensive coverage of the human kinome (including a total of 518 known kinases). All compounds were competitive ATP site-directed inhibitors. On the basis of assays with phosphorylated and unphosphorylated ABL1 kinase, 37 compounds were classified as type I inhibitors and 13 as type II inhibitors[28] (which block the ATP site by binding to different subsites); for 22 compounds no assignment could be made. This classification was preliminary because it was carried out on the basis of only one kinase (and ABL1 mutant forms).[28] Since type 1 and type 2 inhibitors typically display different selectivity profiles,[28] they were both considered for our profile analysis, hence increasing the variety of activity readouts. This also ensured that a sufficient number of compounds were available for machine learning.

For the purpose of our analysis, we then binarized the profiling matrix by designating all compound-kinase interactions for which a measured K_d value was available as "true" (i.e., the compound was considered active against this kinase) and all others as "false" (i.e., the compound was considered inactive). The resulting matrix contained 6,203 "active" and 21,373 "inactive" compound-kinase pairings. Figure 1 shows histograms of enumerated activities per kinase target and compound, respectively. While the majority of targets were inhibited by less than 30 active compounds (Figure 1 a), there were also targets that were inhibited by most of the 72 compounds (e.g., kinase YSK4

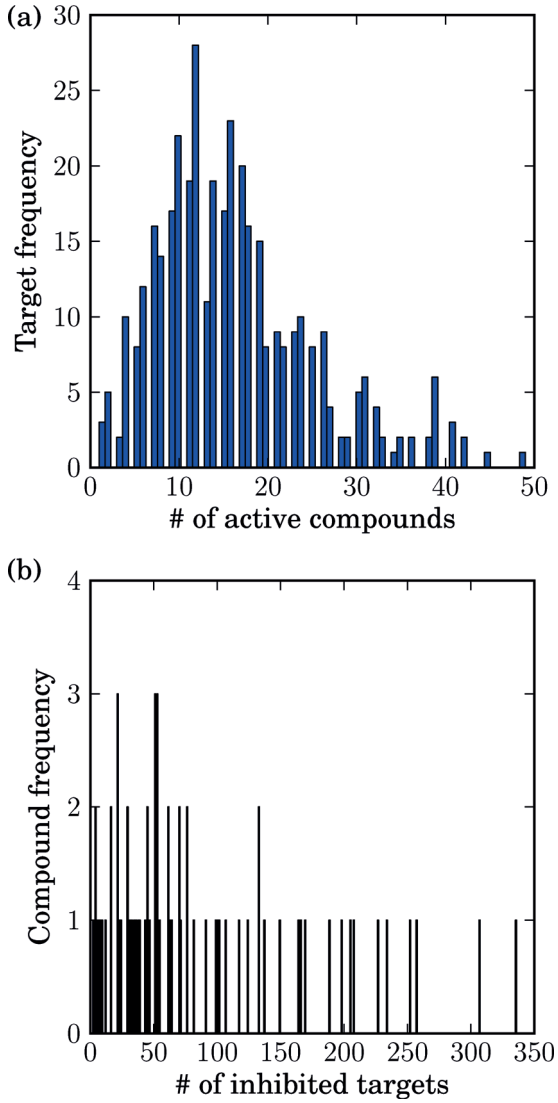


Figure 1. Activity histograms. (a) Number of active compounds per target. Most targets were inhibited by fewer than 30 compounds. (b) Number of inhibited targets per compound. Nearly all inhibitors displayed varying levels of promiscuity; a few compounds inhibited more than 200 kinases.

was inhibited by 49 of the 72 compounds). In addition, the inhibitors displayed a variety of promiscuity patterns (Figure 1 b). Nearly all compounds were active against more than one kinase. The well-known broad spectrum kinase inhibitor staurosporine was active against 336 of all 383 kinases. Notably, each of the 72 compounds exhibited a unique activity profile, i.e., no compounds shared the same activity annotations against all 383 kinases. Pairwise structural and profile similarities of the 72 compounds are provided in Figure S1 of the Supporting Information.

2.2 Naïve Bayesian Classification

In Bayesian classification, a desired class is modeled as a random variable *Y*, which is dependent on a set of observed features **X**.[30] Given a compound represented as a feature vector **x**, we are interested in the conditional probabilities $P(y|\mathbf{x})$ that **x** belongs to each class $y \in$ {active, inactive}. Using Bayes' rule, these *posterior probabilities* can be written as

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \quad (1)$$

Here, $P(\mathbf{x}|y)$ is the *class likelihood* that describes how likely the observation **x** is given a specific class*y*. $P(y)$ is the *prior probability* accounting for the likelihood of class *y* regardless of the observation **x**. In addition, $P(\mathbf{x})$ is called the *evidence*, which is the marginal probability of observation **x**, regardless of the class *y*.[30] A classification is then achieved by choosing the class with the highest posterior probability:

$$\hat{y} = \arg\max_{y \in Y} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \quad (2)$$

In this equation, the evidence serves as a normalization factor to conserve the probability:

$$\sum_{y \in Y} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} = 1 \quad (3)$$

To determine the class with the maximum posterior probability, the evidence does not need to be explicitly determined because the features of any given instance are constant. Thus, to enable the application of the decision rule, only the prior probabilities and the class likelihoods for each class need to be determined:

$$P(y|\mathbf{x}) \propto P(\mathbf{x}|y)\ P(y) \quad (4)$$

The *naïve* Bayesian (NB) classification assumes that all observed features are independent of each other.[31] Even though the feature independence assumption often is an approximation in practice, naïve Bayesian classifiers have shown to perform well on a number of classification tasks, for instance, spam classification.[32,33] Using the independence assumption, the class likelihoods can be written as

$$P(\mathbf{x}|y) = \prod_{j=1}^{d} P(x_j|y) \quad (5)$$

where *d* is the number of input dimensions and $\mathbf{x}_j$ the value of **x** at position *j*.

The prior probability $P(y)$ can be estimated from the training data using *maximum likelihood estimation*.[30] Because one is interested in a binary classification (e.g., 0 = "inactive" or 1 = "active"), it is possible to model *Y* as a Bernoulli random variable:

$$P(y) = p^y\ (1-p)^{1-y} \qquad y \in \{0,1\} \quad (6)$$

Given n training examples, where instance i has class label $y^{(i)}$, the maximum likelihood estimate for p is

$$\hat{p} = \frac{\sum_{i=1}^{n} y^{(i)}}{n} \tag{7}$$

Hence, p can be estimated from the training data as the ratio of active compounds over all compounds in the training set. For the estimation of class likelihoods, individual formulations for different feature representations were used, as further explained below.

2.3 Prediction Models

2.3.1 Structure-Based Classification

For structure-based classification, we represent compounds using structural fingerprints. One example of a (sub)structural fingerprint is MACCS,[34] a binary vector consisting of 166 bits in which each bit encodes the presence or absence of a specific substructure. Hence, in this case, the observations **X** are binary feature vectors. The class likelihoods are estimated from the training data as

$$\hat{p}(x_j|y=1) = \frac{\sum_{i=1}^{n} x_j^{(i)} y^{(i)}}{\sum_{i=1}^{n} y^{(i)}} \tag{8a}$$

$$\hat{p}(x_j|y=0) = \frac{\sum_{i=1}^{n} x_j^{(i)} (1-y^{(i)})}{\sum_{i=1}^{n} (1-y^{(i)})} \tag{8b}$$

That is, the class likelihood of each feature is estimated by the ratio of training samples in the active class that contain this feature.[30]

2.3.2 Profile-Based Classification

In profile-based classification, each compound is "indirectly" represented by its profile, i.e., its known activities against a set of targets. In principle, these profiles also represent binary feature vectors in which each bit indicates if the compound is active or inactive against a given target. However, while features for structure-based classification are always known, this is not necessarily the case for profile-based classification. In practice, a given compound might have only been tested against a subset of targets in a panel. This situation is simulated in our analysis by profiling matrix modifications, as further detailed below. Consequently, an observation x_j in profile-based classification can be assigned to three different states: *active*, *inactive*, or *unknown*. Only two of these states are informative (i.e., active and inactive) and these two can be modeled as Bernoulli features. Hence, we estimate the class likelihood for each feature using a modified version of the maximum likelihood estimate where the unknown state is excluded. Let δ_{ij} be a function that returns 1 if the activity of compound i against target j is known and 0 otherwise. Then we can estimate the class likelihood for the known positions as

$$\hat{p}(x_j|y=1) = \frac{\sum_{i=1}^{n} x_j^{(i)} y^{(i)} \delta_{ij}}{\sum_{i=1}^{n} y^{(i)} \delta_{ij}} \tag{8c}$$

$$\hat{p}(x_j|y=0) = \frac{\sum_{i=1}^{n} x_j^{(i)} (1-y^{(i)}) \delta_{ij}}{\sum_{i=1}^{n} (1-y^{(i)}) \delta_{ij}} \tag{8d}$$

The classification rule is also altered to take only those feature positions into account that are known in the test instance $\mathbf{x}^{(t)}$:

$$\hat{y} = \arg\max_{y \in Y} P(y) \prod_{j=1}^{d} \delta_{tj} P(x_j^{(t)}|y) + (1 - \delta_{tj}) \tag{9}$$

Importantly, both modifications *explicitly exploit the feature independence assumption* of the naïve Bayesian classifier by separating the different features from each other.

2.3.3 Hybrid Classifiers

Instead of taking structural or bioactivity features exclusively into account, we also design *hybrid* naïve Bayesian classifiers. For this purpose, the availability of explicit prior probabilities for each class is exploited (cf. Equation 4). In the first step, class priors are modeled by the maximum likelihood estimation according to Equation 7. Then, either the structure-based or the profile-based classification is applied, as described above. However, instead of choosing the class with the maximum posterior probability, we explicitly calculate the posterior probabilities (see Equation 1). These posteriors then serve as prior probabilities for the next round of classification when the alternative classifier is used (see Figure 2). Hence, in the case of hybrid classifiers, the notion of prior knowledge changes: during the first round, the prior probability of each compound-target interaction represents the estimated general probability that *any* compound is active against a certain target. During subsequent rounds, the prior probability of a compound-target interaction is the predicted posterior probability of another classifier, i.e., the predicted probability that the *given* compound is active against the given target. At a first glance, this might look like a violation of the original NB formulation because the marginal prior probability $P(y)$ is replaced by the conditional probability $P(y|\mathbf{x})$. However, it is important to note that the condition **x** has a different meaning during two subsequent rounds. In structure-based prediction, **x** represents a compound's structure, whereas in profile-based prediction, it represents a compound's activity profile. Having made a prediction $P(y|\mathbf{x}_1)$ in either case, the subsequent one facilitates a prediction $P(y|\mathbf{x}_2)$, which does not involve $\mathbf{x}_1$. Hence, $P(y)$ is conditioned on a variable that is not involved in the subsequent computation, which does not represent a violation of Bayes Theorem.

Furthermore, for each unknown position in the profiling matrix, we derive an individual model that uses the prediction of the previous round as training input for the current

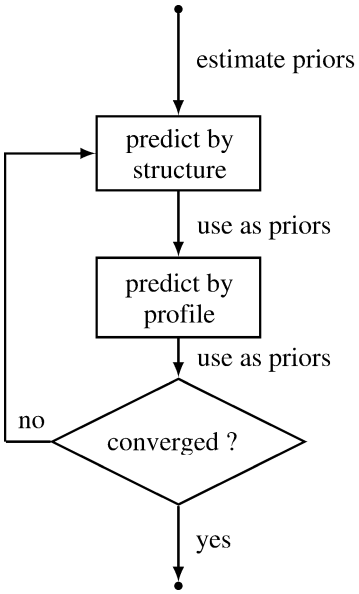


Figure 2. Hybrid classification concept. First, prior probabilities are estimated from the data. The posterior probabilities of compound structure-based prediction are then used as priors for activity profile-based prediction and vice versa (structure-profile classifier). The process is continued until convergence is reached. Alternatively, posterior probabilities of profile-based prediction are first determined and used as priors for structure-based prediction (profile-structure classifier).

classification task. That is, the classifiers of any prediction after the first round can be trained on a complete profiling matrix, even though this complete matrix is the output of another classifier and can therefore only be regarded as an "educated guess". The process of using posterior probabilities as priors for the next prediction continues iteratively until convergence is reached (see Figure 2). As a convergence criterion, predicted class labels of all test instances were required to remain constant for at least three iterations.

2.4 Calculation Set-Up and Performance Assessment

In order to modify the complete profiling matrix and introduce unknown states for prediction, 50% of the compound-target interactions were randomly removed from the data matrix. The remaining 50% of the data were then used to train a structure-based classifier, a profile-based classifier, and two hybrid classifiers, i.e., one beginning with structure-based prediction, termed *structure-profile classifier* and the other beginning with profile-based prediction, termed *profile-structure classifier*. Hence, training compounds for the classifiers had incomplete activity profiles and the number of known active and inactive compounds varied for each target. The so-derived classification models were then used to predict the missing data points in the matrix.

One might also exclude entire rows or columns of the data matrix, which would correspond to activity profile prediction of a formerly untested compound or compound activity prediction against an orphan target, respectively. The former application is not feasible using purely structure-based approaches and the latter is not feasible applying any of the methods explored in this study because no training data for the target model would be available. Nonetheless, these additional predictions should represent interesting opportunities for future work.

As control calculations, we have also carried out structure-based support vector machine (SVM) predictions.[30,35] As rationalized above, activity profile-based classification was not feasible for SVMs, due to the underlying feature independence assumption. SVMs have often produced best results in standard compound activity prediction[36–38] and therefore become a machine learning "gold standard" in chemoinformatics.

Matrix modifications, model derivation, and test calculations were repeated 100 times to obtain statistically meaningful results, which are reported in the following as averages over 100 independent trials.

Structure-based classifiers were derived using MACCS[34] and the extended connectivity fingerprint with bond diameter 4 (ECFP4)[39] as molecular representations. Both fingerprints were calculated using an in-house implementation based upon OpenEye's OEChem toolkit.[40] For MACCS, we used SMARTS patterns adapted from RDKit.[41] For naïve Bayesian and SVM classifiers, the freely available Python implementation scikit-learn was used.[42] For structure-based naïve Bayes classification, the Bernoulli naïve Bayes formulation was applied and the partial model for profile-based naïve Bayes classification was generated with an in-house Python script. The SVM models were built using the Tanimoto kernel[43] with automated class weights. Otherwise, standard parameters were applied to ensure reproducibility of the calculations. That is, we used Laplacian smoothing using $\alpha = 1$ for NB modeling, and support vector classification using $C = 1$.

To assess the performance of the classifiers, *precision*, *recall*, and *balanced/imbalanced accuracy* were calculated. These performance measures are defined as follows (TP: true positive, FP: false positive, TN: true negative, and FN: false negative):

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy}_{\text{balanced}} = 0.5 \cdot \left(\frac{TP}{TP + FN} + \frac{TN}{TP + FP} \right)$$

$$\text{Accuracy}_{\text{imbalanced}} = \frac{TP + TN}{TP + FP + TN + FN}$$

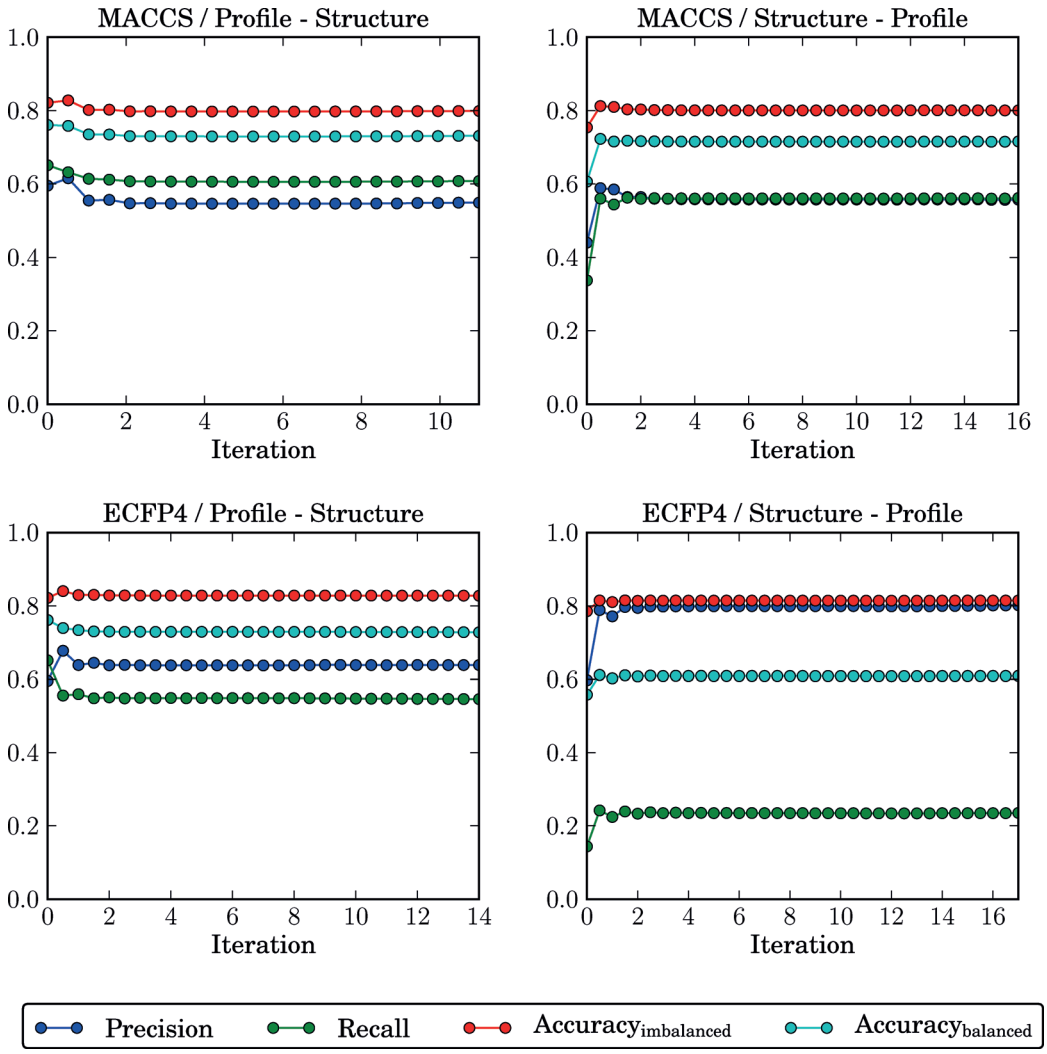


Figure 3. Convergence of hybrid classifiers. Reported is the iterative performance evaluation of structure-profile and profile-structure classification using the MACCS and ECFP4 fingerprints as structural descriptors, respectively. Each iteration is represented by two data points reporting two predictions per round. For both classifiers, the performance level was maximal after the first iteration and then remained essentially constant (structure-profile) or was slightly reduced (profile-structure) until convergence was reached. Standard deviations were consistently below 0.02 and are therefore not reported in this figure.

TP, FP, TN, and FN were only calculated for unknown matrix positions; training positions were not considered. All performance measures range from zero to one. The comparison of imbalanced and balanced accuracy reflects the influence of the imbalanced data composition on the classification results (given, in this case, by the prevalence of negative over positive interactions).

3 Results and Discussion

3.1 Analysis Concept and Aims

A major goal of our study has been the exploration of multi-target activity predictions in high-dimensional target space in the context of compound profiling. Experimental data for such an analysis are currently difficult to obtain. Therefore, we have manually assembled a complete profiling matrix for 72 inhibitors involving nearly 400 related kinase targets. Thus, while the compound set was of limited size, it populated a truly high-dimensional target space. Through iterative random removal of 50% of the activity data from this profiling matrix, a sparsely populated high-dimensional matrix was obtained, hence presenting a challenging prediction task. Through matrix modification it was simulated that profiling was incomplete, i.e., not all compounds were tested against all targets. Hence, predictions of unknown target-compound combinations corresponded to the computational extension of experimental data, a task of practical relevance for chemogenomics. We have explored both compound structure-based activity prediction, the conventional approach in chemoinformatics, and activity profile-based prediction, which is much less ex-

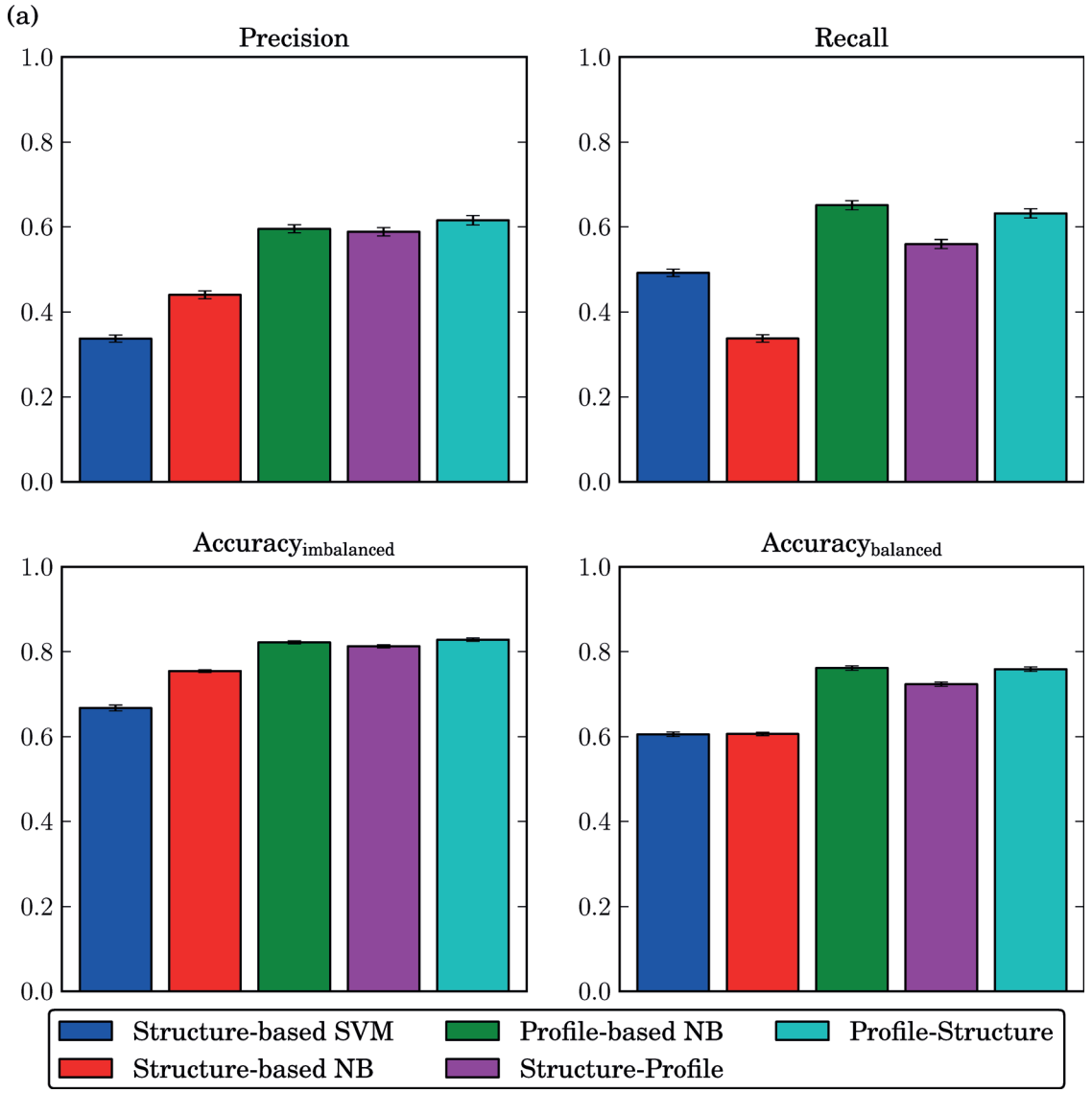


Figure 4. Performance of NB classifiers. Mean and standard deviations of precision, recall, imbalanced, and balanced accuracy are reported for all classification models using (a) MACCS and (b) ECFP4 as descriptors for structure-based classification.

plored. Profile-based prediction, as carried out herein, is related to the experimental concept of affinity fingerprinting,[21] which has been applied in biological screening to small target sets. Profile-based classification requires the prediction of the activity of a compound against a novel target on the basis of activity data available for other closely or distantly related targets. Hence, profile-based prediction does not involve consideration of compound structure. In addition to structure- and profile-based classification, hybrid prediction models were designed taking both structure- and profile information into account. These models were more complex than individual classifiers, but would in principle be expected to maximize prediction performance as they utilized all available information.

In the following, the performance of the different types of classifiers is compared and an in-depth feature analysis is provided. We begin with analyzing the convergence characteristics and prediction performance of the hybrid models, given their new design.

3.2 Convergence and Performance of Hybrid Classifiers

Hybrid classification models were derived with two alternative molecular representations. Using the MACCS fingerprint, hybrid classifiers converged on average after 11.1 iterations in profile-structure and after 16.6 iterations in structure-profile prediction, with standard deviations (*SD*) of 4.9 and 6.4 iterations, respectively. Using ECFP4, the profile-structure classifier converged after 14.3 iterations (*SD* = 6.3),

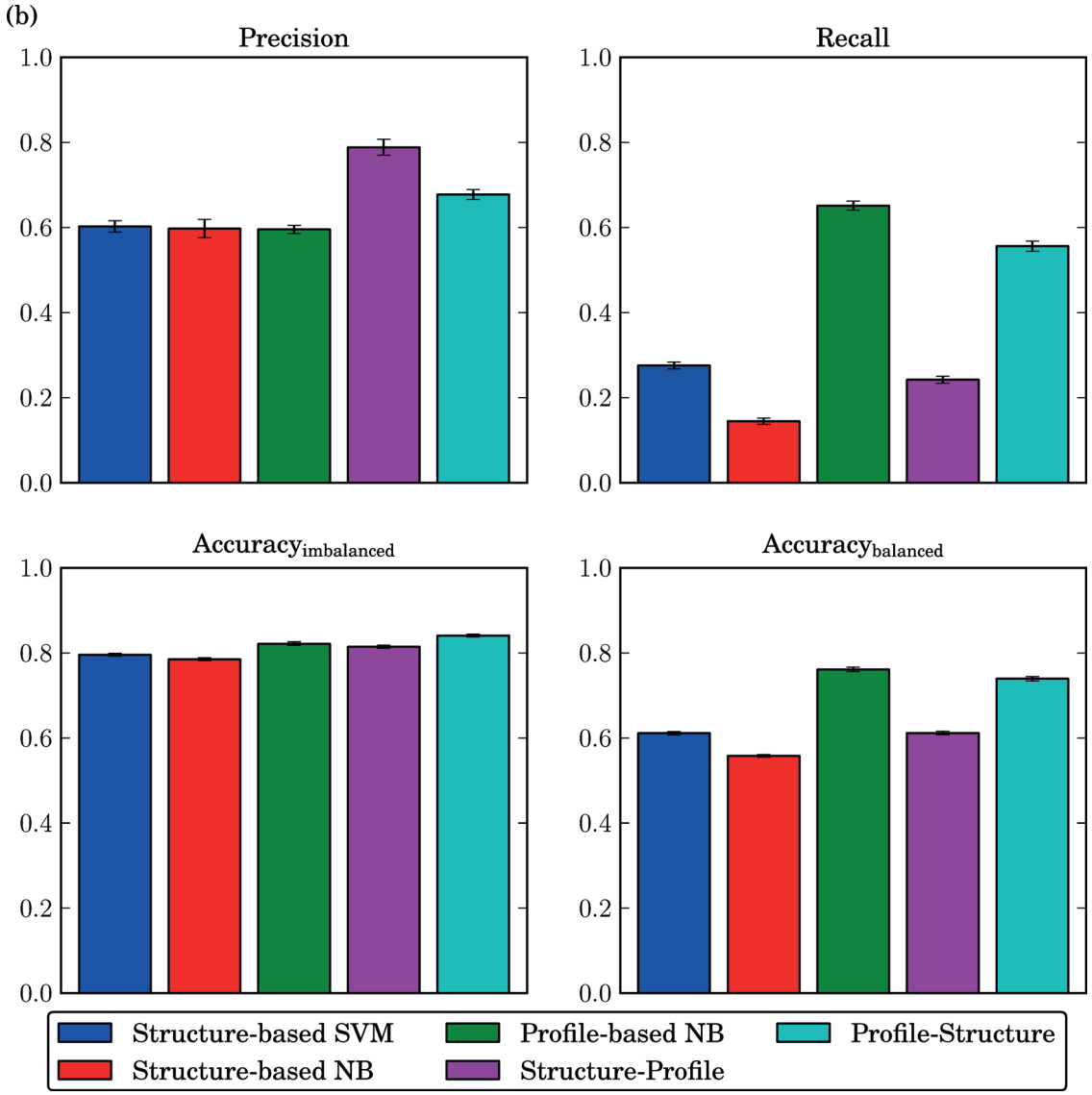


Figure 4b

and the structure-profile classifier after 17.7 iterations (*SD* = 6.2). Six individual trials of the structure-profile and 11 trials of the profile-structure classification using ECFP4 did not converge during the first 100 iterations and were discontinued.

Figure 3 reports different performance measures monitoring iterative classification. Interestingly, the general trend was observed that performance was essentially maximal after the initial iteration and did not notably improve during subsequent rounds; a rather unexpected finding. Both structure-profile and profile-structure models appear to be dominated by the profile-based prediction; this can be inferred from the finding that following the first data point in profile-structure and the second data point in structure-profile prediction, no further improvement is observed. It is likely that the posterior probabilities following profile-based prediction dominated the priors during the next round such that no further changes were observed.

Hence, at least for our data set, iterative learning with inferred class labels did not provide an advantage over single-step classification, indicating that the results were stable after initial conditional probability assignment. Therefore, in the following, we consider the performance of the hybrid classifiers after the initial iteration.

3.3 Performance Comparison of Alternative Classification Models

Figure 4 reports the mean and SD of the prediction performance of all classifiers including structure-based SVM control calculations using MACCS (Figure 4a) and ECFP4 (Figure 4b) as descriptors. First, the results are discussed for MACCS. In this case, the SVM classifier produced the lowest

precision with only 33.72%, but had higher recall (49.23%) than the structure-based NB classifier. Accordingly, its imbalanced accuracy was lowest with 66.77% and the balanced accuracy was comparable to the structure-based NB classifier (60.55%). The structure-based NB classifier generally predicted fewer active positions, which resulted in higher precision (44.04%) and lower recall (33.73%) than the SVM classifier. Accordingly, the imbalanced accuracy of the structure-based NB classifier was higher (by 8.66%), because negative positions were overrepresented in the data.

Strikingly, all classifiers using activity profile information for prediction outperformed structural classifiers by all performance measures. Moreover, the structure-profile classifier produced lower performance (58.86% precision, 56.01% recall, 81.28% imbalanced, and 72.32% balanced accuracy) than the profile-based NB and the profile-structure classifier. Both the profile-structure and the profile-based NB classifier produced comparable results, with precision of 61.56% and 59.53%, recall of 63.19% and 65.11%, imbalanced accuracy of 82.83% and 82.18%, and balanced accuracy of 75.86% and 76.12%, respectively. Hence, overall promising predictions were obtained. Furthermore, these findings revealed that profile-based prediction was superior to structure-based prediction in the case of the high-dimensional kinase profiling matrix and that overall best performance was achieved by single-step profile-based classification.

Since the MACCS fingerprint consists of 166 structural features, while 382 activity features were available for profile-based learning, we also investigated the higher-dimensional ECFP4 atom environment fingerprint[39] for structure-based classification. This fingerprint does not have a constant format, but generates varying numbers of topological features in a molecule-specific manner.

Figure 4b shows the performance of the ECFP4-based structural classifiers compared to profile-based and hybrid classifiers. In this case, both SVM and NB classifications displayed a significant increase in precision, i.e., 26.49% to 60.21% for SVM and 15.66% to 59.70% for NB, but the recall was substantially reduced to 27.55% and 14.44%, respectively. This means that the structure-based classifiers predicted fewer active instances with ECFP4 than with MACCS. Hence, imbalanced accuracy was further increased, whereas balanced accuracy was reduced for the ECFP4 structure-based NB classifier (while it remained approximately at the same level for ECFP4-based SVM classification). A similar picture emerged for structure-profile prediction. Here, precision was improved by 20% to 78.86% and recall reduced from 56.01% to 24.19%. Therefore, balanced accuracy also decreased nearly to the level of the structure-based classifiers. However, profile-structure prediction suffered significantly less from these effects, i.e., precision was only increased by 6.17% and recall was decreased by only 7.62%. Taken together, the results revealed an expected fingerprint dependence of structure-based classification, but also showed that increasing fingerprint feature space did not improve these predictions.

Figure S4 of the Supporting Information reports the performance of the different classifiers from a more profile-centric perspective. Here, the number of individual profiles is plotted against their number of incorrect positions. Using structure-based SVM and MACCS, on average, at least 20 positions were incorrectly predicted in each profile. Half of all 72 profiles could be predicted with a maximum error of 50 positions and for all profiles, the average maximum error was 100 interactions. Profile-based NB predictions were able to generate profiles with only one incorrect position, and half and all of the profiles could be predicted with a maximum error of 21 and 103 positions, respectively. On the other hand, when the structural representation was changed to ECFP4, the SVM classifier was able to predict profiles with only four false positions. Predicting half and all of the compound profiles could be done with a mean error of 24 and 119 positions, respectively.

Overall, profile-based activity predictions were clearly superior to structure-based classification in the case of our high-dimensional kinase data set. In fact, the incorporation of structural information in hybrid models partly compromised profile-based activity prediction. Thus, in this case, the dominance of profile-based prediction in hybrid classifiers was not a conceptual flaw, but desirable.

The results indicate that profile-based predictions provide a viable complement and/or alternative to ligand-based approaches in cases where ligand data is sparse but a sufficient number of biological annotations is available. Hence, for high-dimensional activity spaces, profile-based modeling should be a highly attractive approach. In cases where large numbers of compounds are available, it should be interesting to compare profile- and ligand-based predictions in detail. We would expect that they might be complementary dependent on the compound classes under study. For practical applications, an advantage of profile-based modeling as reported herein is the fact that no structure information is required at all to achieve accurate predictions, which circumvents the generally observed compound class dependence of ligand-based methods and supports transferability of the approach to high-dimensional activity spaces of different target composition. Given that ligand data was sparse in our case, it should also be noted that different estimation strategies exist to handle sparse ligand-target data and complement available data with computational extrapolations.[44] This points at another attractive feature of profile-based predictions because in this case, it was not required to further extend ligand data.

3.4 Feature Analysis

In order to explore possible reasons for the superior performance of profile-based predictions, classification models were analyzed in detail and it was attempted to identify features leading to accurate predictions. Initially, we investigated which targets were most influenced by replacing structural with profile descriptors. As a reference, the

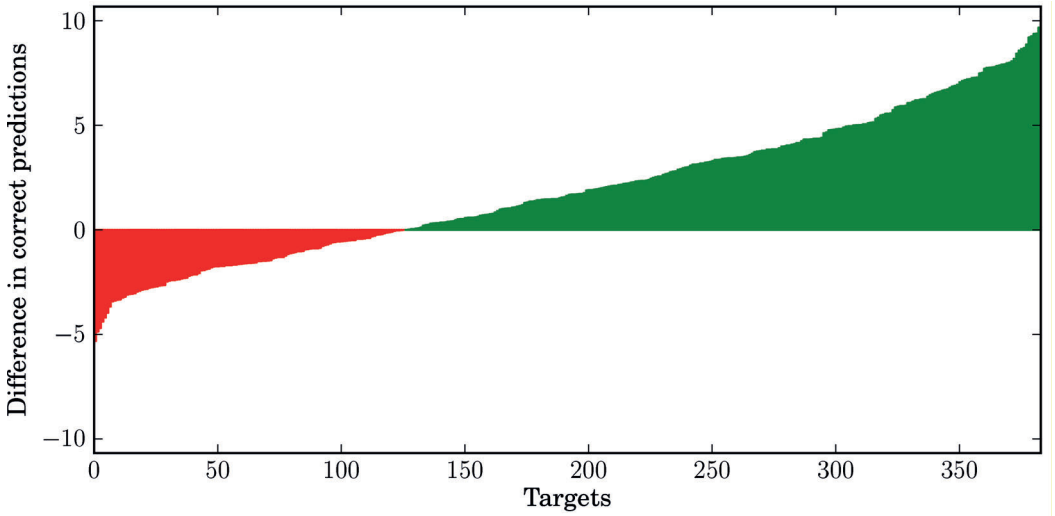


Figure 5. Difference between true prediction rates. Reported is the difference between true (correct) predictions of profile- and structure-based NB classification. Correct predictions were averaged over all trials. For clarity, targets were sorted according to the observed difference. For approximately 1/3 of the targets, structure-based classification produced, on average, up to five more true predictions (red). By contrast, for approx. 2/3 of the targets, profile-based classification improved the performance over structure-based prediction by, on average, up to 10 individual activity predictions (green).

MACCS-based NB classifier was used because it produced higher balanced accuracy than the ECFP4-based classifier.

Figure 5 reports the average difference in correctly predicted compound activities per target for the structure- and the profile-based NB classifier. The per-target results rationalize the observed global performance difference, as discussed above. For 125 targets, structure-based classification yielded more correct predictions (with up to, on average, 27.14 correctly predicted positions vs. 21.83 for the profile-based classifier), whereas more correct predictions were obtained by profile-based classification for 257 targets (up to, on average, 24.01 correctly predicted positions vs. 14.33 for the structure-based classifier). Only one target had the same number of correct predictions for both classifiers. The five kinase targets with largest gain in positive predictions in profile-based classification and the five targets with largest loss are listed in Table 1. In the following, the profile-based prediction models for the kinase with the largest performance improvement (Yamaguchi sarcoma viral oncogene homolog 1; YES) and the largest performance reduction (mitogen-activated protein kinase 4; ERK4) are analyzed.

Table 1. Targets with largest improvement and reduction in prediction performance. Reported are averaged unknown positions and true predictions for the top five targets with largest improvement and the top five targets with largest reduction in prediction performance by profile- compared to structure-based classification.

Target	Unknown positions	True predictions (profile-based)	True predictions (structure-based)
Top 5 targets with largest performance improvement			
YES	28.41	24.01 (84.51 %)	14.33 (50.44 %)
TNIK	27.82	21.57 (77.53 %)	12.18 (43.78 %)
MAP3K2	28.51	25.26 (88.60 %)	15.89 (55.73 %)
FYN	28.75	24.11 (83.86 %)	14.85 (51.65 %)
JAK2	27.91	26.16 (93.73 %)	17.3 (61.98 %)
Top 5 targets with largest performance reduction			
ERK4	27.84	21.83 (78.41 %)	27.14 (97.49 %)
TRPM6	27.96	22.74 (81.33 %)	27.61 (98.75 %)
ERK1	28.07	22.83 (81.33 %)	27.53 (98.08 %)
VRK2	29.44	24.3 (82.54 %)	28.69 (97.45 %)
ERK2	28.34	23.66 (83.49 %)	27.86 (98.31 %)

3.4.1 Target with Largest Performance Improvement

On average, 43.59 reference compounds were utilized for YES, 19.94 of which were active and 23.65 inactive. Figure 6 compares the experimentally observed activities of the 72 compounds against YES with the results of structure- and profile-based predictions. Shading indicates the percentage of all trials in which a given compound was predicted to be active or inactive. The structure-based classifier yielded a larger number of shaded profile positions indicating that the predictions were less stable than for the profile-based classifier. Table 2 gives the average number of true and false positive and negative predictions for the two classifiers. It is evident that the profile-based classifier produced substantially fewer false positive, but more true negative predictions than the structure-based classifier. To better understand this performance ratio, the inhibitor crizotinib is considered (Figure 7). The compound was active against YES and used as a test instance in 34 of the 100 trials. It was predicted to be active by the profile-based classifier in all 34 trials, whereas the structure-based classifier only predicted it to be active in two trials. Considering the meth-

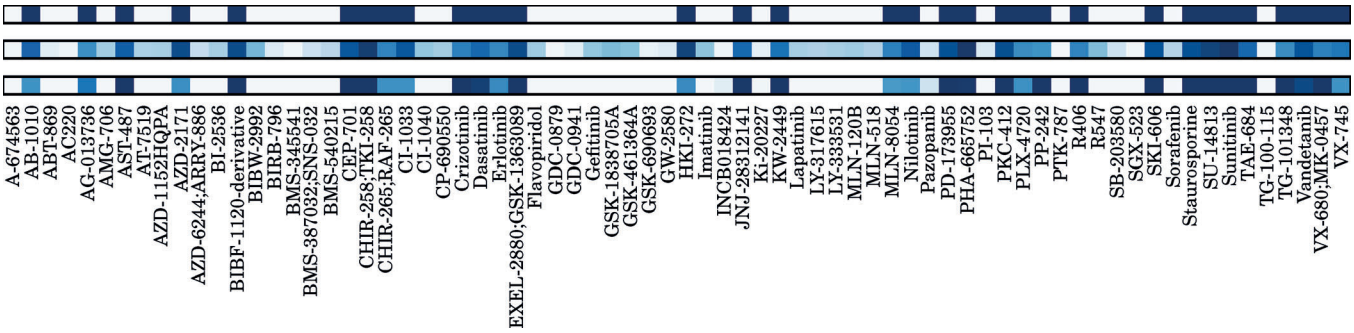


Figure 6. Predicted vs. observed compound activities for kinase YES. The profile at the top shows the experimentally determined activities of the 72 compounds against YES. Profile positions are color-coded by observed inhibition: dark blue indicates inhibition and white no inhibition. The middle and bottom profiles show averaged activity predictions from the structure- and profile-based classification, respectively. A dark blue or white position indicates that a compound was predicted to be active or inactive in all (or nearly all) of the independent trials. Blue shading of profile positions indicates the ratio of active vs. inactive predictions; i.e. the darker the blue shade, the more active predictions were observed over all trials.

Table 2. True and false predictions for kinase YES. For YES, average numbers of true (Active/Active, Inactive/Inactive) and false (Active/Inactive, Inactive/Active) predictions are reported for structure- and profile-based classification.

Experiment	Structure-based prediction		Profile-based prediction	
	Active	Inactive	Active	Inactive
Active	26.59	6.41	28.83	4.17
Inactive	7.67	31.33	0.23	38.77

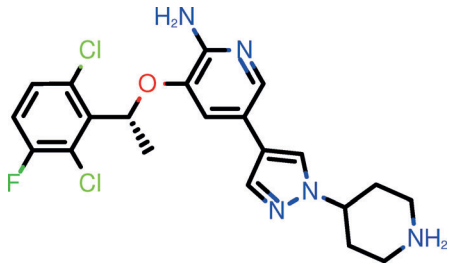


Figure 7. Crizotinib. The structure of this exemplary inhibitor is shown (with conventional atom coloring) whose activity against kinase YES was consistently correctly predicted in profile-based, but not structure-based classification.

odological basis for the predictions, as discussed above, the following criterion had to be met for an active prediction (cf. Equation 9):

$$P(\text{active})\prod_{j=1}^{d}\delta_{tj}P(x_j^{(t)}|\text{active})+(1-\delta_{tj}) > P(\text{inactive})\prod_{j=1}^{d}\delta_{tj}P(x_j^{(t)}|\text{inactive})+(1-\delta_{tj}) \quad (10)$$

The equation can be rearranged as follows:

$$\prod_{j=1}^{d}\frac{\delta_{tj}P(x_j^{(t)}|\text{active})+(1-\delta_{tj})}{\delta_{tj}P(x_j^{(t)}|\text{inactive})+(1-\delta_{tj})} > \frac{P(\text{inactive})}{P(\text{active})} \quad (11)$$

Table 3. Targets with largest influence on the crizotinib/YES classification. Listed are kinase targets with the highest ratio $P(x_j|\text{active})/P(x_j|\text{inactive})$ for test compound crizotinib and the profile-based classification model for target YES. Crizotinib was experimentally active against all listed targets. For each target, average conditional feature probabilities are reported.

Target	x_j	$P(x_j\|\text{inactive})$	$P(x_j\|\text{active})$
SYK	active	0.0665	0.4108
FGR	active	0.1651	0.9072
EPHA3	active	0.1063	0.6061
TXK	active	0.0826	0.4802
ITK	active	0.0707	0.4678
SIK2	active	0.0964	0.6580
AMPK-α1	active	0.0625	0.4988
MAP4K3	active	0.0923	0.6823
DLK	active	0.0633	0.5085
LZK	active	0.0614	0.5441

For a given target, prior probabilities for both class labels are constant and the δ_{tj} functions are constant for the same compound and prediction model. It follows that we need to analyze the features x_j yielding the highest ratio $P(x_j|\text{active})/P(x_j|\text{inactive})$ to rationalize a classification result.

This ratio of conditional feature probabilities was averaged over the 34 trials in which crizotinib was a test compound and the kinases with the highest ratio are listed in Table 3. Interestingly, only targets against which crizotinib was also active significantly influenced the active classification of crizotinib for YES. For example, it was known from the training data that crizotinib was active against SYK. The NB classifier then utilized this information by comparing the conditional probabilities for the observation "activity against SYK", given the possible cases "active against YES" and "inactive against YES". The remaining conditional probabilities $P(\text{SYK}=\text{inactive}\,|\,\text{YES}=\text{active})$ and $P(\text{SYK}=\text{inactive}\,|\,\text{YES}=\text{inactive})$ were not considered because the observation "SYK = inactive" was not true for crizotinib.

The conditional probability for crizotinib to be active against SYK, given the possible case that it was also active against YES, was given by $P(\text{SYK}=\text{active}\,|\,\text{YES}=\text{active})=0.4108$. On the other hand, the conditional probability for crizotinib to be active against SYK, given the possible case of inactivity against YES, was given by $P(\text{SYK}=\text{active}\,|\,\text{YES}=\text{inactive})=0.0665$. Hence, the observation that crizotinib was active against SYK was, on the basis of the NB model, due to activity of crizotinib against YES (41.08%) with much higher probability than due to inactivity (6.65%). Thus, the fact that crizotinib was active against SYK played a major role for the prediction of this compound as active against YES. Analogous considerations apply to the other targets listed in Table 3. For example, the conditional probabilities of crizotinib to be active against FGR (which had a lower $P(x_j|\text{active})/P(x_j|\text{inactive})$ ratio than SYK), given inactivity/activity against YES, were 16.51% and 90.72%, respectively.

Furthermore, to explain the correlation between conditional feature probabilities and correct predictions not only for a given compound, but all predictions for YES (thus generalizing the analysis for a given target), we have distinguished between true positive and true negative predictions. For each true positive prediction in each trial, targets were selected that met the condition $P(x_j|\text{inactive})/P(x_j|\text{active})\geq 2$, i.e., only those targets were taken into account that had at least a two times higher probability to (correctly) vote for the active than (incorrectly) for the inactive class. Analogously, for each true negative prediction, targets were selected meeting the condition $P(x_j|\text{inactive})/P(x_j|\text{active})\geq 2$. Moreover, we have distinguished the case where $x_j=\text{active}$ and $x_j=\text{inactive}$ to account for all possible conditional probabilities involving x_j and y and the states active and inactive.

On the basis of the above criteria, only targets with the highest influence on correct predictions were considered, depending on their own inhibition by a specific compound. Targets meeting these criteria in at least 50% of all trials are listed in Table 4. Because there were 889 true positive predictions for YES over all trials, a selected target had to meet the above criteria at least 445 times. Analogously, a target selected to have a significant influence on negative predictions had to meet the above criteria for at least 756 of all 1512 true negative predictions.

The results in Table 4 show that only targets against which a given compound was active strongly influenced true positive predictions for YES, and only target against which a compound was inactive strongly influenced true negative predictions. Hence, the presence of target-activity correlation effects played a major role for the quality of the predictions. While most of the targets having a significant influence on correct predictions belonged to the same subfamily as YES (i.e., subfamily TK), kinases from other subfamilies such as STE and CAMK were also found. There were 15 targets influencing true positive predictions, given they were also inhibited, but only five targets influencing true negative predictions, given they were not inhibited. Interestingly, these five targets including FGR, FYN, and SRC from the TK subfamily, MAP4K3 from STE, and SIK2 from CAMK, also had a significant influence on true positive predictions.

Table 4. Kinases with strong influence on true predictions for target YES. Listed are kinases with significance influence on individual true positive and true negative predictions for target YES and the subfamilies[28] they belong to. These targets were required to have an at least two times higher conditional probability for correct than incorrect predictions in at least 50% of all test cases (see text for details).

Target	Subfamily	x_j	Prediction
AXL	TK	active	active
BLK	TK	active	active
FGFR2	TK	active	active
FGR	TK	active	active
FYN	TK	active	active
HPK1	STE	active	active
JAK2	TK	active	active
JAK3	TK	active	active
MAP3K2	STE	active	active
MAP4K2	STE	active	active
MAP4K3	STE	active	active
MERTK	TK	active	active
SIK2	CAMK	active	active
SRC	TK	active	active
TNIK	STE	active	active
FGR	TK	inactive	inactive
FYN	TK	inactive	inactive
MAP4K3	STE	inactive	inactive
SIK2	CAMK	inactive	inactive
SRC	TK	inactive	inactive

3.4.2 Target with Largest Performance Reduction

ERK4 was the kinase with the largest reduction in true predictions for profile-based compared to structure-based classification. On average, 27.14 compound activities were correctly predicted using structure-based classification in this case, but only 21.83 compound activities using profile-based classification. Figure 8 compares the experimentally observed activities of the 72 compounds against ERK4 with the results of structure- and profile-based predictions and Table 5 reports the average number of true and false positive and negative predictions for the two classifiers. ERK4 was inhibited by only two compounds in the data set, i.e., erlotinib and PD-173955. Both classifiers were able to correctly predict these two interactions in most of the trials; however, in this case, the profile-based classifier produced more false positives.

In accordance with our analysis on YES, false positive and false negative predictions for ERK4 in all trials were considered. For each false positive prediction, targets were selected meeting the condition $P(x_j|\text{inactive})/P(x_j|\text{active})\geq 2$, and for each false negative prediction, targets were select-

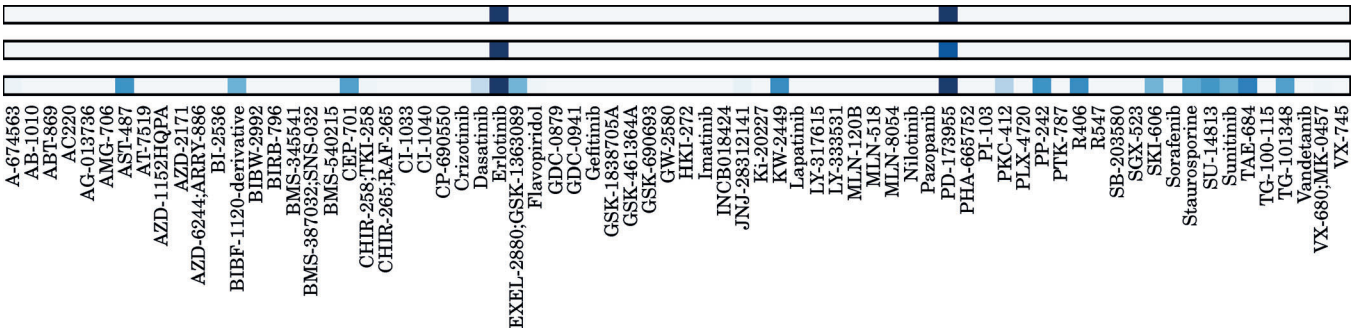


Figure 8. Predicted vs. observed compound activities for kinase ERK4. Experimentally determined (top) and predicted (middle: structure-, bottom: profile-based) compound activities are shown for ERK4. The representation is according to Figure 6.

Table 5. True and false predictions for kinase ERK4. For ERK4, average numbers of true (Active/Active, Inactive/Inactive) and false (Active/Inactive, Inactive/Active) predictions are reported for structure- and profile-based classification.

Experiment	Structure-based prediction		Profile-based prediction	
	Active	Inactive	Active	Inactive
Active	1.30	0.70	1.37	0.63
Inactive	0.00	70.00	5.38	64.62

ed meeting the condition $P(x_j|\text{inactive})/P(x_j|\text{active}) \geq 2$. Hence, the selected targets had the strongest influence on incorrect predictions. Kinases meeting the above criteria in at least 50% of the trials are listed in Table 6. Here, false negative predictions were detected in only 63 of all possible instances but false positive predictions were detected 538 times.

Again, only targets that were also inhibited had a notable influence on "active" predictions and only targets that were not inhibited influenced "inactive" predictions. We identified a total of 21 kinases with an incorrect influence on positive predictions (most of them belonging to the TK subfamily, whereas ERK4 belonged to CMGC). However, no target from the CMGC subfamily was found to compromise prediction of ERK4. There was only one target (AXL) that influenced both false positive and false negative predictions.

To put our analysis into perspective, we have also calculated pairwise Tanimoto similarities of the targets with respect to their compound activities. The so obtained *Tc* similarity values were mostly low, as reported in Figure S2 of the Supporting Information. For YES, *Tc* values ranged from 0.0 to 0.86, with a mean of 0.32. For ERK4, *Tc* values between 0.0 and 0.29 with a mean of 0.03 were obtained (Figure S3 of the Supporting Information). In addition, Tables S1 and S2 of the Supporting Information list the *Tc* values of the most influential targets listed in Tables 4 and 6, respectively. The *Tc* values of the most influential targets for YES ranged from 0.58 to 0.86 and were thus relatively high. However, there were also other targets having a high *Tc* relative to YES that were not influential (e.g., LYN with a *Tc* of 0.73). On the other hand, *Tc* values of ERK4 and its influential targets ranged from 0.0 to 0.08, which means that none of the targets with high *Tc* influenced the performance of the ERK4 model in a negative way. Hence, these findings might indicate that sharing the same inhibitors is a necessary, but insufficient condition for a target to strongly influence profile-based NB predictions.

Table 6. Targets with significant influence on false predictions for target ERK4. Listed are kinases with significance influence on individual false positive and false negative predictions for target ERK4.

Target	Subfamily	x_j	Prediction
ALK	TK	active	active
AXL	TK	active	active
EPHB1	TK	active	active
FER	TK	active	active
FES	TK	active	active
FGFR3	TK	active	active
FRK	TK	active	active
INSR	TK	active	active
JAK1	TK	active	active
LIMK2	TKL	active	active
MAP3K15	STE	active	active
MAP3K3	STE	active	active
MAP3K4	STE	active	active
MAP3K2	STE	active	active
PRKX	AGC	active	active
SGK3	AGC	active	active
SIK	CAMK	active	active
SNARK	CAMK	active	active
SYK	TK	active	active
TLK2	OTHER	active	active
ULK2	OTHER	active	active
AXL	TK	inactive	inactive
LTK	TK	inactive	inactive
MEK4	STE	inactive	inactive
PDGFRA	TK	inactive	inactive
SIK2	CAMK	inactive	inactive

4 Concluding Remarks

In this work, we have carried out machine learning-based activity predictions in high-dimensional target space with

the aid of compound profiling data, focusing on inhibition of nearly 400 protein kinases. Predictions were carried out to complete experimental observations in a profiling matrix by assigning activity states of unknown compound-target combinations. This corresponds to a computational extension of experimental profiling data, which is of practical relevance, especially for chemogenomics applications. The presence of incomplete profiling data principally challenges conventional machine learning approaches for class label prediction. Therefore, different prediction approaches were explored including compound structure- and activity profile-based classification as well as hybrid methods taking structure and profile information into account. Profile-based prediction conceptually benefited from the application of a feature independence assumption and utilization of three possible activity states (i.e., active, inactive, and unknown). Therefore, naïve Bayesian classification was chosen as the method of choice for prediction and complemented with support vector machine control calculations. It was anticipated that iterative hybrid classification utilizing structure and profile information might provide the best basis to maximize prediction performance. However, activity profile-based NB classifiers yielded overall more accurate predictions than hybrid methods or SVM or NB classifiers using molecular representations. Thus, consideration of compound structure was not required for accurate activity prediction in high-dimensional target space. Moreover, its inclusion was even unfavorable in a number of cases. This observation was likely due to the often high degree of structural similarity of the ATP site-directed kinase inhibitors under study. Of course, one needs to consider that compound structure information was by default limited and that structure-based predictions might also be further improved once more compounds become publicly available that have been extensively profiled against large numbers of targets. On the other hand, using more compounds for model building would also further increase the number of available training profiles and thus support profile-based prediction. Taken together, our results clearly emphasize the value of profile information for activity prediction in high-dimensional target space, at least in the context of Bayesian classification. Interesting questions for follow-up work include, for example, to what extent these findings are influenced by the similarity of ligands and the degree of correlation between activity profiles. It should also be of interest to explore high-dimensional data sets in which ligand and profile data are balanced (that are, unfortunately, difficult to obtain at present). It is reasonable to assume that in the presence of balanced and comparably informative structural and profiling data, hybrid classification methods would give an advantage over both individual approaches. Hence, although hybrid classification was inferior to profile-based predictions in our current study, hybrid methods continue to merit further exploration, for principal reasons.

We have also attempted to rationalize the findings through a detailed analysis of two exemplary profile-based prediction models, which revealed that profile-based predictions were driven by implicitly learned target-activity correlations. These correlation effects were not limited to kinases belonging to the same subfamily but also involved kinases from different subfamilies. Such correlation effects favor activity profile-based predictions and are effectively exploited through the derivation of conditional probabilities of activity in the context of naïve Bayesian classification. Hence, in high-dimensional target spaces, such as the kinase space investigated herein, activity prediction on the basis of profile information is found to be a promising approach and can be effectively applied by learning from incomplete profiling data. In addition, increasing compound coverage of high-dimensional target space might also further increase the performance of the hybrid classification models introduced herein. Structures, molecular fingerprints, and ChEMBL IDs of the compounds investigated herein are made freely available.[45]

Conflict of Interests

No conflict of interests declared.

Acknowledgements

The authors thank *Martin Vogt* for helpful discussions on probabilities and naïve Bayesian classification. The OEChem toolkit was provided by *OpenEye's* free academic licensing program.

References

[1] J. A. Allen, B. L. Roth, *Annu. Rev. Pharmacol. Toxicol.* **2011**, *51*, 117 – 144.
[2] M. A. Fabian, W. H. Biggs III, D. K. Treiber, C. E. Atteridge, M. D. Azimioara, M. G. Benedetti, T. A. Carter, P. Ciceri, P. T. Edeen, M. Floyd, J. M. Ford, M. Galvin, J. L. Gerlach, R. M. Grotzfeld, S. Herrgard, D. E. Insko, M. A. Insko, A. G. Lai, J. M. Lélias, S. A. Mehta, Z. V. Milanov, A. M. Velasco, L. M. Wodicka, H. K. Patel, P. P. Zarrinkar, D. J. Lockhart, *Nature Biotechnol.* **2005**, *23*, 329 – 336.
[3] D. M. Goldstein, N. S. Gray, P. P. Zarrinkar, *Nature Rev. Drug. Discov.* **2008**, *6*, 391 – 397.
[4] J. T. Metz, E. F. Johnson, N. B. Soni, P. J. Merta, L. Kifle, P. J. Hajduk, *Nature Chem. Biol.* **2011**, *7*, 200 – 202.
[5] F. Milletti, J. C. Hermann, *ACS Med. Chem. Lett.* **2012**, *3*, 383 – 386.
[6] D. Rognan, *Br. J. Pharmacol.* **2007**, *152*, 38 – 52.
[7] Z. A. Knight, H. Lin, K. M. Shokat, *Nat. Rev. Cancer* **2010**, *10*, 130 – 137.
[8] D. Dimova, P. Iyer, M. Vogt, F. Totzke, M. H. G. Kubbutat, C. Schächtele, S. Laufer, J. Bajorath, *J. Med. Chem.* **2012**, *55*, 11067 – 11071.
[9] L. Jacob, J.-P. Vert, *Bioinformatics* **2008**, *24*, 2149 – 2156.

[10] J. Bajorath, *Mol. Inf.* **2013**, *32*, 1025 – 1028.
[11] A. L. Hopkins, *Nature Chem. Biol.* **2008**, *4*, 682 – 690.
[12] M. J. Keiser, V. Setola, C. Laggner, A. I. Abbas, S. J. Hufeisen, N. H. Jensen, M. B. Kuijer, R. C. Matos, T. B. Tran, R. Whaley, R. A. Glennon, J. Hert, K. L. Thomas, D. D. Edwards, B. K. Shoichet, B. L. Roth, *Nature* **2009**, *462*, 175 – 181.
[13] E. Lounkine, M. J. Keiser, S. Whitebread, D. Mikhailov, J. Hamon, J. L. Jenkins, P. Lavan, E. Weber, A. K. Doak, S. Côté, B. K. Shoichet, L. Urban, *Nature* **2012**, *486*, 361 – 367.
[14] C. R. Chong, D. J. Sullivan, *Nature* **2007**, *448*, 645 – 646.
[15] A. Shiraishi, S. Niijima, J. B. Brown, M. Nakatsui, Y. Okuno *J. Chem. Inf. Model.* **2013**, *53*, 1251 – 1262.
[16] Z. Simon, Á. Peragovics, M. Vigh-Smeller, G. Csukly, L. Tombor, Z. Yang, G. Zahoránszky-Köhalmi, L. Végner, B. Jelinek, P. Hári, C. HeténYi, I. Bitter, P. Czobor, A. Málnási-Csizmadia, *J. Chem. Inf. Model.* **2012**, *52*, 134 – 145.
[17] Á. Peragovics, Z. Simon, I. Brandhuber, B. Jelinek, P. Hári, C. HeténYi, P. Czobor, A. Málnási-Csizmadia, *J. Chem. Inf. Model.* **2012**, *52*, 1733 – 1744.
[18] E. Martin, P. Mukherjee, *J. Chem. Inf. Model.* **2012**, *52*, 156 – 170.
[19] S. Niijima, A. Shiraishi, Y. Okuno, *J. Chem. Inf. Model.* **2012**, *52*, 901 – 912.
[20] J. Balfer, K. Heikamp, S. Laufer, J. Bajorath, *Chem. Biol. Drug Des.* **2014**, *83*, DOI: 10.1111/cbdd.12294, in press.
[21] L. M. Kauvar, D. L. Higgins, H. O. Villar, J. R. Sportsman, Å. Engqvist-Goldstein, R. Bukar, K. E. Bauer, H. Dilley, D. M. Rocke, *Chem. Biol.* **1995**, *2*, 107 – 118.
[22] A. F. Fliri, W. T. Loging, P. F. Thadeio, R. A. Volkmann, *J. Med. Chem.* **2005**, *48*, 6918 – 6925.
[23] A. M. Wassermann, E. Lounkine, M. Glick, *J. Chem. Inf. Model.* **2013**, *53*, 692 – 703.
[24] P. M. Petrone, B. Simms, F. Nigsch, E. Lounkine, P. Kutchukian, A. Cornett, Z. Deng, J. W. Davies, J. L. Jenkins, M. Glick, *ACS Chem. Biol.* **2012**, *7*, 1399 – 1409.
[25] A. Bender, J. L. Jenkins, M. Glick, Z. Deng, J. H. Nettles, J. W. Davies, *J. Chem. Inf. Model.* **2006**, *46*, 2445 – 2456.
[26] DiscoveRx Corporation, 42501 Albrae Str., Fremont, CA 94538, USA.
[27] http://www.discoverx.com/tools-resources/interaction-maps.
[28] M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber, P. P. Zarrinkar, *Nat. Biotechnol.* **2011**, *29*, 1046 – 1052.
[29] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, J. P. Overington, *Nucleic Acids Res.* **2012**, *40*, D1100 – D1107.
[30] E. Alpaydin, *Introduction to Machine Learning,* 2nd ed., MIT Press, Cambridge, USA, **2010**.
[31] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification,* 2nd ed., Wiley-Interscience, New York, **2000**.
[32] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, "A Bayesian Approach to Filtering Junk Email", *AAAI Workshop on Learning for Text Categorization,* Madison, Wisconsin, **1998**.
[33] H. Zhang, "The Optimality of Naïve Bayes", *Proc. 17th Int. Florida Artific. Intell. Res. Soc. Conf.* **2004**, pp 562–567.
[34] *MACCS Structural Keys*, Accelrys, San Diego, CA, **2011**.
[35] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed., Springer, New York, **2000**.
[36] K. Kawai, S. Fujishima, Y. Takahashi, *J. Chem. Inf. Model.* **2008**, *48*, 1152 – 1160.
[37] K. Heikamp, J. Bajorath, *J. Chem. Inf. Model.* **2013**, *53*, 791 – 801.
[38] J. Meslamani, R. Bhajun, F. Martz, D. Rognan, *J. Chem. Inf. Model.* **2013**, *53*, 2322 – 2333.
[39] D. Rogers, M. Hahn, *J. Chem. Inf. Model* **2010**, *50*, 742 – 754.
[40] *OEChem TK*, version 2.0.0, OpenEye Scientific Software, Santa Fe, NM; http://www.eyesopen.com.
[41] *RDKit*, Open-source cheminformatics; http://www.rdkit.org/.
[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, *J. Mach. Learn. Res.* **2011**, *12*, 2825 – 2830.
[43] L. Ralaivola, S. J. Swamidass, H. Saigo, P. Baldi, *Neural Netw.* **2005**, *18*, 1093 – 1110.
[44] Y. Tanrikulu, R. Kondru, G. Schneider, W. V. So, H.-M. Bitter, *Mol. Inf.* **2010**, *29*, 678 – 684.
[45] J. Balfer, Y. Hu, J. Bajorath, ZENODO DOI:10.5281/zenodo.9223.

Received: April 5, 2014
Accepted: May 20, 2014
Published online: July 25, 2014

Summary

This chapter covered the application of a naïve Bayes classification approach to incomplete activity profiling data, and a detailed analysis of the resulting model. It was shown that in high-dimensional target space, good prediction accuracy can be achieved without taking compound structure information into account. Furthermore, the design of the profile-based classifier exploited the naïve assumption of feature independence to enable the application to incomplete profiling data. This contribution is especially important in practice, since publicly available chemogenomics data is seldomly complete and usually only sparsely distributed. Moreover, the resulting models can be used for an analysis of the target space at hand, in this case, the human kinome.

In the analysis, we showed which targets benefit most from the structure-independent prediction, and which related kinases had most influence on compound-target interaction classification. As such, our analysis provides two main insights into computational activity profile modeling: First, in the presence of high-dimensional profiling data, it has to be carefully investigated whether structural or profiling information should be used for the prediction of compound-target interactions. Second, our approach makes it possible to not only apply profile-based predictions to incomplete data, but it also provides means to interpret how similar related targets are to each other in terms of compound binding.

The following chapter deals with the task of compound activity prediction against single targets. Instead of classifying active and inactive compounds, SVR is used to model the potency values of presumably active compounds. Again, the focus is not on the method application and benchmarking, but on understanding the mechanisms that lead to success or failure of the applied models.

4

Systematic Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis

Introduction

While the previous chapters dealt with the classification of compound-target interactions as active or inactive, the next study performs a potency regression task. Here, it is assumed that all compounds in a given data set are active at different levels, and their potency is to be predicted. This task is usually applied in the lead optimization stage, where several active compounds have already been identified. Models are then built for these compounds, with the aim of ranking them in order of their estimated potency. For this task, SVR is one of the most popular models in the chemoinformatics community. Failure or success of these models are usually measured statistically, in terms of R^2 scores, squared or absolute errors.

In this study, we build SVR models for a variety of compound data sets with known activity information. These models are then used to predict the potency values of previously unseen compounds, and their statistical performance in terms of R^2 scores and mean absolute errors is derived. Furthermore, quantitative measures of SAR continuity and discontinuity, as well as qualitative representations such as the SAR landscape, are utilized. Our results show that even though the global prediction accuracy of the models was at least acceptable and often good, the most interesting SAR regions are systematically mispredicted.

PLOS ONE

RESEARCH ARTICLE

Systematic Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis

Jenny Balfer, Jürgen Bajorath*

Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Dahlmannstr. 2, D-53113, Bonn, Germany

* bajorath@bit.uni-bonn.de

OPEN ACCESS

Citation: Balfer J, Bajorath J (2015) Systematic Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis. PLoS ONE 10(3): e0119301. doi:10.1371/journal.pone.0119301

Academic Editor: Andrea Cavalli, University of Bologna & Italian Institute of Technology, ITALY

Received: December 18, 2014

Accepted: January 29, 2015

Published: March 5, 2015

Data Availability Statement: All data have been made freely available via the public ZENODO repository. DOI: 10.5281/zenodo.13986 (http://dx.doi.org/10.5281/zenodo.13986).

Funding: The authors received no specific funding for this work.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Support vector machines are a popular machine learning method for many classification tasks in biology and chemistry. In addition, the support vector regression (SVR) variant is widely used for numerical property predictions. In chemoinformatics and pharmaceutical research, SVR has become the probably most popular approach for modeling of non-linear structure-activity relationships (SARs) and predicting compound potency values. Herein, we have systematically generated and analyzed SVR prediction models for a variety of compound data sets with different SAR characteristics. Although these SVR models were accurate on the basis of global prediction statistics and not prone to overfitting, they were found to consistently mispredict highly potent compounds. Hence, in regions of local SAR discontinuity, SVR prediction models displayed clear limitations. Compared to observed activity landscapes of compound data sets, landscapes generated on the basis of SVR potency predictions were partly flattened and activity cliff information was lost. Taken together, these findings have implications for practical SVR applications. In particular, prospective SVR-based potency predictions should be considered with caution because artificially low predictions are very likely for highly potent candidate compounds, the most important prediction targets.

Introduction

Support vector machines (SVMs) are algorithms for supervised machine learning [1] that have become increasingly popular for object classification and ranking in bioinformatics [2,3] and chemoinformatics [4,5], given their often observed high predictive performance compared to other machine learning approaches [5]. The basic idea underlying SVM modeling is to derive classification models by separating positive and negative training data with the largest possible margin. Furthermore, SVMs are often used in combination with kernel functions, which project training sets into feature spaces of higher dimensionality where a linear separation of

positive and negative training data might ultimately be feasible. The resulting models are then used to predict test instances.

In addition to classification and ranking, the SVM approach has also been adapted for prediction of numerical property values through support vector regression (SVR) [6,7]. Instead of constructing a hyperplane for classification, SVR derives a function on the basis of training data to predict numerical values. SVR is an intrinsically non-linear prediction approach because it projects data sets characterized by the presence of non-linear structure-property relationships in original feature spaces into higher-dimensional space representations where a linear regression function can be fitted. Accordingly, SVR has been receiving much attention in recent years in the context of quantitative structure-activity relationship analysis (QSAR) [8] to predict activities of bioactive compounds. QSAR has been, and continues to be, the most widely applied computational approach for potency prediction and compound design in medicinal chemistry.

Classical QSAR modeling attempts to predict changes in compound potency that result from small chemical modifications using linear regression models [8]. Therefore, these predictions are typically limited to series of structural analogs in which the assumption of at least approximate linearity of structure-activity relationships (SARs) holds. By contrast, the prediction of potency values of compounds from large and structurally heterogeneous data sets, in which SARs are typically non-linear, fall outside the applicability domain of classical QSAR and require non-linear prediction methods such as neural networks [8] or SVR. In addition to potency prediction [9–11], SVR has also been applied to predict a variety of other compound-associated property values [12–16]. SVR models derived for potency prediction reported in the literature are typically statistically assessed and cross-validated following standard QSAR procedures, i.e., by calculating coefficients of determination to account for the ability of a model to fit the potency values of the training data and predict test data not utilized for model building [8].

In this work, we have carried out an in-depth performance evaluation of SVR models for potency prediction beyond standard statistics. SVR models were derived for a variety of data sets with different SAR characteristics and systematically analyzed for their ability to predict compound potency values and vulnerability to over- or underfitting potency data. Furthermore, SVR model regularization terms were systematically varied to balance model complexity and permitted training errors in different ways. For all data sets, activity landscapes [17] were generated from experimental measurements and compared to landscape representations derived on the basis of SVR predictions. Although SVR models were generally statistically sound due to accurate predictions of many intermediate potency values, the models were affected by underfitting and consistently inaccurate predictions of the most potent compounds, leading to a smoothing effect on modeled activity landscapes and loss of critical SAR information.

Materials and Methods

Compound data selection

From the public database ChEMBL [18], release 17, all sets of compounds active against human targets were selected that contained at least 500 molecules. Furthermore, qualifying compounds were required to be experimentally tested in a direct inhibition or binding assay with highest ChEMBL confidence score. Only equilibrium constants (K_i values) below 100 μM were considered, hence omitting weakly active compounds and assay-dependent measurements. Multiple K_i values available for the same compound were averaged if they fell into the same order of magnitude; otherwise, the compound was discarded. Furthermore, duplicates, known pan-assay interference compounds [19], and other reactive molecules were removed

Table 1. Data overview.

TID	Target name	no. of cpds.	min.pKi	max.pKi	mean pKi
11	Thrombin	654	5.00	12.19	6.78
15	Carbonic anhydrase II	1221	5.00	9.41	6.97
51	Serotonin 1a (5-HT1a) receptor	1342	5.05	10.85	7.74
72	Dopamine D2 receptor	1791	5.00	10.24	6.97
87	Cannabinoid CB1 receptor	1661	5.00	10.10	6.92
100	Norepinephrine transporter	928	5.03	9.66	6.94
107	Serotonin 2a (5-HT2a) receptor	824	5.01	11.00	7.54
108	Serotonin 2c (5-HT2c) receptor	577	5.00	9.70	7.01
114	Adenosine A1 receptor	1911	5.01	10.52	6.61
121	Serotonin transporter	1229	5.02	10.89	7.44
129	Mu opioid receptor	1504	5.01	11.80	7.45
130	Dopamine D3 receptor	1142	5.05	10.05	7.37
136	Delta opioid receptor	1203	5.01	10.60	7.19
137	Kappa opioid receptor	1399	5.02	11.52	7.50
138	Nociceptin receptor	642	5.04	10.70	7.84
155	Dopamine transporter	745	5.04	9.80	6.70
165	HERG	701	5.00	9.26	6.14
176	Purinergic receptor P2Y12	536	5.36	9.40	7.81
194	Coagulation factor X	1129	5.02	11.40	8.05
252	Adenosine A2a receptor	2189	5.01	11.09	6.91
259	Cannabinoid CB2 receptor	1841	5.00	10.40	7.17
278	Adenosine A2b receptor	856	5.05	9.80	7.31
280	Adenosine A3 receptor	1766	5.02	10.56	7.20
10142	Melanocortin receptor 4	1199	5.01	9.40	6.96
10193	Carbonic anhydrase I	1134	5.00	10.68	6.33
10280	Histamine H3 receptor	1861	5.04	10.50	7.95
10627	Serotonin 6 (5-HT6) receptor	1157	5.06	10.30	7.76
11290	Histamine H4 receptor	596	5.04	10.40	7.08
12209	Carbonic anhydrase XII	717	5.00	9.52	7.24
12952	Carbonic anhydrase IX	1033	5.01	9.92	7.07
19905	Melanin-concentrating hormone receptor 1	701	5.03	9.77	7.42

For all 31 data sets used for SVR modeling, the ChEMBL target ID (TID), target name, number of compounds, and the minimum, maximum, and mean pK_i values are reported.

doi:10.1371/journal.pone.0119301.t001

from all data sets using in-house computational filters. On the basis of these stringent selection criteria, 31 compounds sets with activities against diverse targets were obtained for SVR modeling, as summarized in Table 1. All data sets are freely available for download from the public ZENODO platform [20].

Molecular representations

For all test compounds, two fingerprints were calculated as descriptors including molecular access system (MACCS) keys [21] and the extended connectivity fingerprint with bond diameter 4 (ECFP4) [22]. MACCS is a fixed-length fingerprint consisting of 166 pre-defined substructural patterns and ECFP4 a topological atom environment fingerprint of higher chemical resolution. ECFP generates all possible atom environments up to a layer of four bonds around each

atom. The resulting atom environments represent a feature set of data-specific size. Both fingerprint representations were calculated using in-house implementations based upon OpenEye's OEChem toolkit [23].

SAR information content

The SAR characteristics of the 31 compound data sets used for SVR modeling were quantitatively described using the continuity and discontinuity score components of the SAR Index (SARI) [24]. For all data sets, initial scores were calculated as follows:

$$\mathrm{cont}_{\mathrm{raw}} = 1 - \frac{\sum_{i>j} w_{ij} Tc(i,j)}{\sum_{i>j} w_{ij}} \quad (1)$$

$$\mathrm{disc}_{\mathrm{raw}} = \frac{\sum_{\{i,j|Tc(i,j)\geq t, i>j\}} |pot(i) - pot(j)| Tc(i,j)}{|\{i,j|Tc(i,j) \geq t, i > j\}|} \quad (2)$$

Here, the *Tc(i,j)* is the Tanimoto coefficient [25,26] of ligands *i* and *j*, w_{ij} is a weight defined as $\frac{pot(i)pot(j)}{1+|pot(i)-pot(j)|}$, and *pot(i)* is the potency of compound *i* as pKi value [24]. For the discontinuity score, we used corresponding Tc similarity threshold values of $t = 0.85$ for MACCS and $t = 0.56$ for ECFP4 [27]. Raw scores were then converted into Z-scores and normalized using the cumulative normal distribution. For this purpose, 120 compound data sets containing at least 100 compounds each were extracted from ChEMBL on the basis of the selection criteria specified above and used as an external reference panel.

Activity landscapes

In chemoinformatics and medicinal chemistry, activity landscapes of compound data sets are generally defined as graphical representations that integrate molecular similarity and activity relationships between compounds [17]. In addition to numerical SAR characterization, three-dimensional (3D) activity landscape views [28] were calculated for all compound data sets subjected to SVR modeling. The 3D activity landscape representation was calculated as described previously based upon a two-dimensional (2D) projection of all pairwise Tc values for a compound set using multi-dimensional scaling [28]. This projection provides a 2D similarity map that is then complemented by an activity surface interpolated from compound potency values. The potency surface is then added to the projection, yielding a 3D activity landscape representation reminiscent of geographical landscape views.

Support vector regression modeling

Support vector regression theory. Support vector regression (SVR) is a supervised machine learning method for the prediction of numerical target values [6,7]. For training, labeled examples are mapped into a descriptor space and a function of the form

$$f(x) = \langle w, x \rangle + b \quad (3)$$

is derived that best predicts the target values for the examples *x*. The parameters *w*,*b* are

derived via the following optimization:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$
$$\text{subject to} \begin{cases} y_i - \langle w, x_i\rangle - b & \leq \epsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases} \quad (4)$$

This concept is derived from support vector machines (SVMs), which were introduced for binary classification tasks. While SVMs attempt to maximize the margin between two classes, SVR derives a so-called *ε-insensitive tube* around the target values [7]. The width of this tube, ε, provides the amount of permitted error, i.e., target values that are mispredicted by less than ε are not penalized by the optimization. Training examples that are predicted with a deviation of more than ε from their true target value fall outside the tube and are called *support vectors*.

Furthermore, ξ_i ξ_i^* in formula (4) are sets of non-negative *slack variables* permitting a certain violation of the ε-tube's bounds [29]. The *regularization term C* balances the cost of a complex model with the cost of training errors: if *C* is large, training errors are strongly penalized and the derived model is highly complex, thus entailing a risk of data *overfitting*. In contrast, if the regularization term is small, low-complexity models are favored at the risk of *underfitting*. In the linear case, the vector *w* can be written as a weighted combination of the support vectors:

$$w = \sum_i (\alpha_i - \alpha_i^*)x_i \quad (5)$$

Hence, the prediction function can be expressed as:

$$f(x) = \sum_i (\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b \quad (6)$$

Importantly, if a linear regression modeling of the training data in the given space is not possible, the scalar product $\langle\cdot,\cdot\rangle$ can be replaced by a kernel function *K(u,v)* to project the data into higher dimensional space in which a linear separation becomes feasible, in analogy to SVMs. This procedure is generally referred to as the *kernel trick* [30]. If the kernel trick is applied, the weight vector *w* can no longer be directly expressed and the prediction function changes to:

$$f(x) = \sum_i (\alpha_i - \alpha_i^*)K(x_i, x) + b \quad (7)$$

In analogy to Tc-based similarity calculations, the *Tanimoto kernel* [31] is often used as a kernel function for compound potency prediction:

$$K(u, v) = \frac{\langle u, v\rangle}{\langle u, u\rangle + \langle v, v\rangle - \langle u, v\rangle} \quad (8)$$

Performance analysis. To assess the overall performance of the SVR modeling, we have calculated absolute errors, mean absolute errors, and R^2 values. According to equation (9), the

absolute error was defined as given by the objective function:

$$\sum_{i=1}^{n}(\xi_i + \xi_i^*) = \sum_{i=1}^{n}\max(0, |y_i - f(x_i)| - \epsilon) \quad (9)$$

While this is not the standard formula for the absolute error, it reflects the training of the models with a certain amount of permitted deviation. This value was used to determine the best regularization term for each model, provided by the value for *C* that resulted in the lowest absolute error on the test set.

The mean absolute error is given as the absolute error divided by *n*, the number of compounds. Furthermore, coefficients of determination (R^2) values were computed. R^2 quantifies how much of the data variance can be explained by the model.

Calculation set-up

For each data set, 55 different SVR models were trained with *C* varying in {1, 2, 3. . ., 50, 100, 250, 500, 1000}. Hence, these models covered a wide range of regularization parameters enabling the analysis of potential under- and overfitting effects. All SVR models were derived using the freely available Python implementation scikit-learn [32] with parameter setting $\varepsilon = 0.1$. We have kept the parameter ε constant instead of optimizing it considering the nature of the data. Because target values were pKi values (which are well-defined) setting ε to values smaller than 0.1 would be beyond experimental detection limits. Moreover, variations of larger values are also not meaningful because deviations of close to one order of magnitude or more are biologically relevant and should not be treated as allowed deviations. For each of the 55 different *C* settings, 10 models were built with randomly chosen training and test sets, each comprising of 50% of the data (yielding a total of 550 models per set). In each case, prediction performance was averaged over all 10 independently derived models.

Results and Discussion

Regularization analysis

Initially, the effects of regularization term variations on SVR model performance were evaluated in detail. Therefore, the mean training and test errors over all individual trials were determined for each setting of *C*. Because this regularization parameter balances model complexity and permitted training errors, its variation makes it possible to elucidate the tendency of over- or underfitting of the models. For small values of *C*, both training and test errors are typically high, which then provides a clear indication of underfitting. Increasing values of *C* should then lead to a decrease in training and test error, indicating a better fit of a model. The application of increasingly large values of *C* values typically leads to increasing test errors in the presence of constant or further decreased training errors, which indicates overfitting of a model.

On the basis of these general considerations, the value of *C* yielding the smallest test error was selected as the preferred regularization term for each data set / fingerprint combination. Table 2 reports the preferred values for all data sets and fingerprints. The performance of the resulting models is further discussed in the following. It should be noted that in benchmark studies, parameters are typically evaluated on an external validation set, while performance values are reported on a distinct test set. However, in this study, we have not aimed to benchmark SVR models and deliberately introduced a positive bias towards model performance to emphasize the consistently observed failure in correctly predicting the most important compounds, as further discussed below. In our regularization analysis, consistent trends were observed across all data sets. Fig. 1 reports the mean training and test errors under regularization term variation

Table 2. Best regularization values.

TID	MACCS	ECFP4
11	23	8
15	10	3
51	17	3
72	7	5
87	8	3
100	6	2
107	7	3
108	7	3
114	11	4
121	9	4
129	16	7
130	9	7
136	12	4
137	14	3
138	6	3
155	7	3
165	9	3
176	12	2
194	11	5
252	14	4
259	9	4
278	9	3
280	27	6
10142	18	6
10193	11	2
10280	14	4
10627	12	3
11290	16	5
12209	16	5
12952	14	4
19905	5	6

Reported are the values of the regularization parameter *C* yielding the lowest test error for each data set and fingerprint representation.

doi:10.1371/journal.pone.0119301.t002

for an exemplary data set (melanocortin receptor 4 antagonists, TID 10142), which mirrors the observed trends. First, both training and test errors were consistently higher for MACCS- than for ECFP4-based models. As a consequence, effects of underfitting and, in part, overfitting depending on the choice of *C* were clearly observed for MACCS but to a much lesser extent, if at all, for ECFP4. The training error of the ECFP4 model reached its minimum of 0.0165 already at $C = 19$, while the training error of the MACCS model continued to steadily decrease for increasing values of *C*. However, for both models, the test error essentially remained constant over most regularization term settings. Only for the highest *C* values, an overfitting effect became apparent for MACCS. This was in contrast to the ECFP4 model that did not reveal an apparent overfit at any point. Moreover, regardless of the data set and molecular representation used, the difference between training and test errors was consistently large. These results

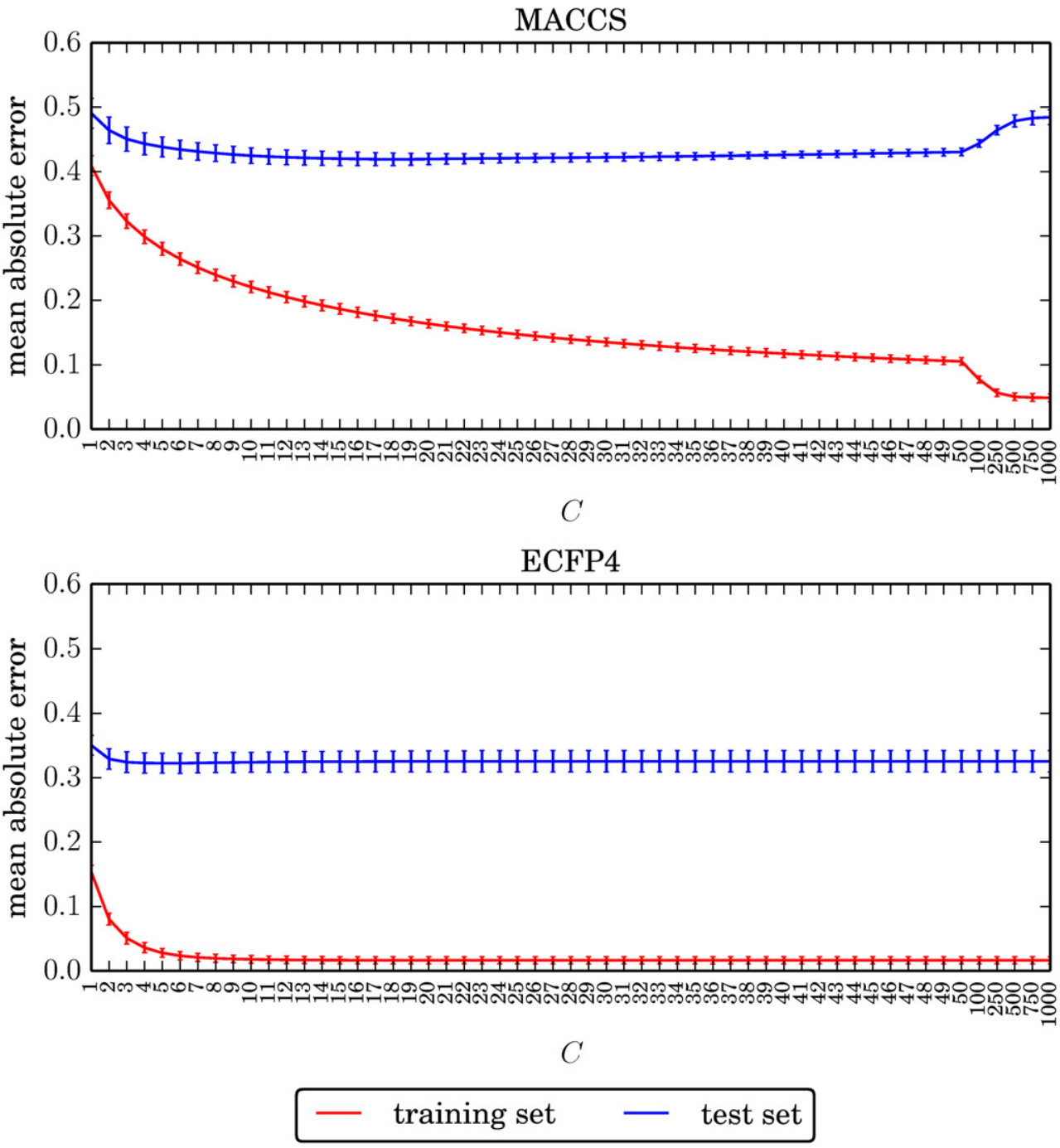


Fig 1. Exemplary regularization profile. For each value of the regularization term *C*, the absolute training and test error averaged over all trials is reported for data set TID 10280. Error bars give the standard deviations. Regularization values are plotted evenly on the x axis, regardless of their magnitude.

doi:10.1371/journal.pone.0119301.g001

revealed that the SVR models in combination with a suitable high-dimensional representation (i.e., ECFP4 instead of MACCS) did not display a notable tendency of overfitting, which often severely affects QSAR modeling [8]. By contrast, underfitting of SVR models was observed for low regularization terms.

Regression performance

We next determined global regression accuracy of the SVR models on the basis of R^2 values and mean absolute errors calculated on the test set. Therefore, for each data set, the overall best performing model identifed under systematic variation of regularization parameter *C* was selected. R^2 values and mean absolute errors of these models are shown in Fig. 2. For all data sets, the performance using MACCS was lower than for the higher-resolution topological ECFP4 fingerprint. MACCS-based models also required generally higher regularization term values than ECFP4-based models (Table 2). Overall, R^2 values ranged from 0.31 and 0.65 (MACCS) and from 0.44 and 0.75 (ECFP4). Hence, these values were of moderate magnitude. However, mean prediction errors were generally low and fell into the pK_i value intervals [0.32, 0.57] and [0.26, 0.50] for MACCS and ECFP4, respectively. Thus, compound potencies were generally predicted well within an order of magnitude, which is considered encouraging accuracy from a QSAR perspective. Of course, the most interesting test compounds for prediction across large data sets were those with high potency. However, prediction results for individual compounds cannot be rationalized on the basis of average errors. Therefore, alternative measures were applied to evaluate the quality of the SVR predictions for potent compounds, as discussed in the following.

SAR characteristics and predictive performance

SAR continuity and discontinuity scores were calculated for all data sets to characterize their global SAR information and relate these SAR characteristics to SVR model performance. High discontinuity scores indicate the presence of many structurally similar compounds with large potency variations, whereas high continuity scores account for the presence of structurally similar or dissimilar compounds with small to moderate variations in potency [24]. Activity cliffs, which consist of pairs or groups of structurally analogous compounds with largest differences in potency, represent the extreme form of SAR discontinuity in a data set [33]. They also represent the most prominent and informative features of activity landscapes [17]. In large and structurally heterogeneous data sets, as investigated herein, continuous and discontinuous SAR environments typically coexist and determine the global SAR phenotype [17,24].

To evaluate if differences in SAR characteristics influenced the fit of SVR models, the correlation of continuity / discontinuity scores and test errors was assessed. To render test errors independent of data set size, the mean error over all test compounds was determined and the Pearson product-moment correlation coefficient between SAR scores and the mean error was calculated. In addition, a two-tailed p-value was determined to assess the statistical significance of correlation effects. These calculations revealed no notable correlation between the SAR continuity score and the prediction error, but a statistically significant correlation ($p < 0.01$) between the discontinuity score and the prediction error. The corresponding Pearson correlation coefficient was 0.75 for MACCS and 0.61 for ECFP4, which indicated the presence of a moderate positive correlation between the discontinuity scores and the mean test error. Thus, the more discontinuous a data set was at the global SAR level, the higher was the prediction error. In Fig. 3, the discontinuity scores of all data sets are plotted against their mean test errors, which reflects this trend.

SVR-based reproducibility of SAR characteristics

Discontinuity scores based on predicted potency values. To further evaluate the findings discussed above, we recalculated continuity and discontinuity scores on the basis of potency values predicted for all training and test set compounds. The *predicted training set* continuity / discontinuity was calculated on the basis of potency predictions for training set compounds

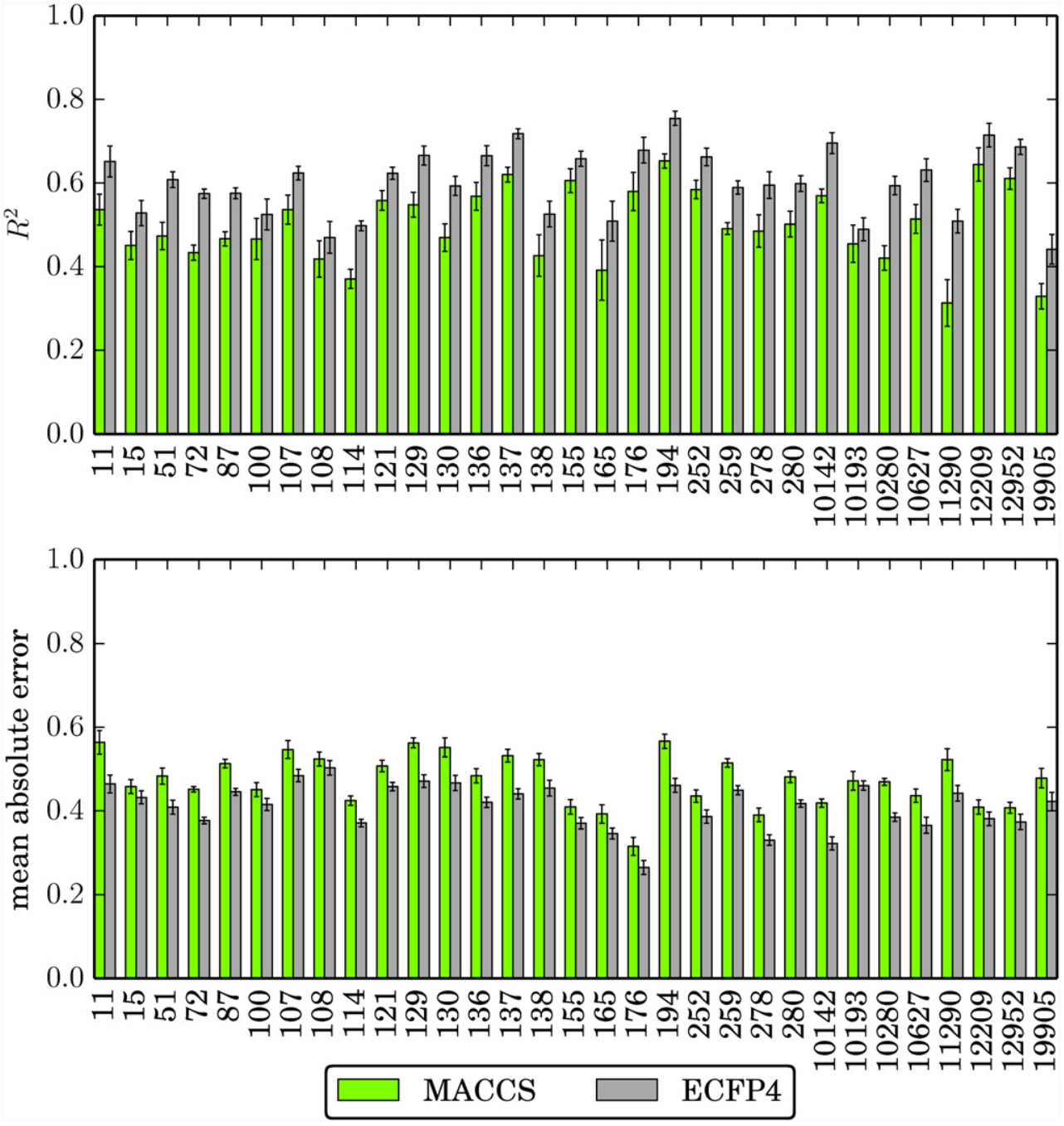


Fig 2. Global regression performance. Reported are the mean R^2 and mean absolute error values for each data set and fingerprint, determined on the test set. Error bars give the standard deviations.

doi:10.1371/journal.pone.0119301.g002

and provides an estimate of the influence of the training error and model fit on SAR characteristics. In contrast, the *predicted test set* continuity / discontinuity was calculated based on test set potency predictions and reflects the generalization ability of the model and its influence on prediction accuracy. Furthermore, in the following, *observed* continuity / discontinuity refers

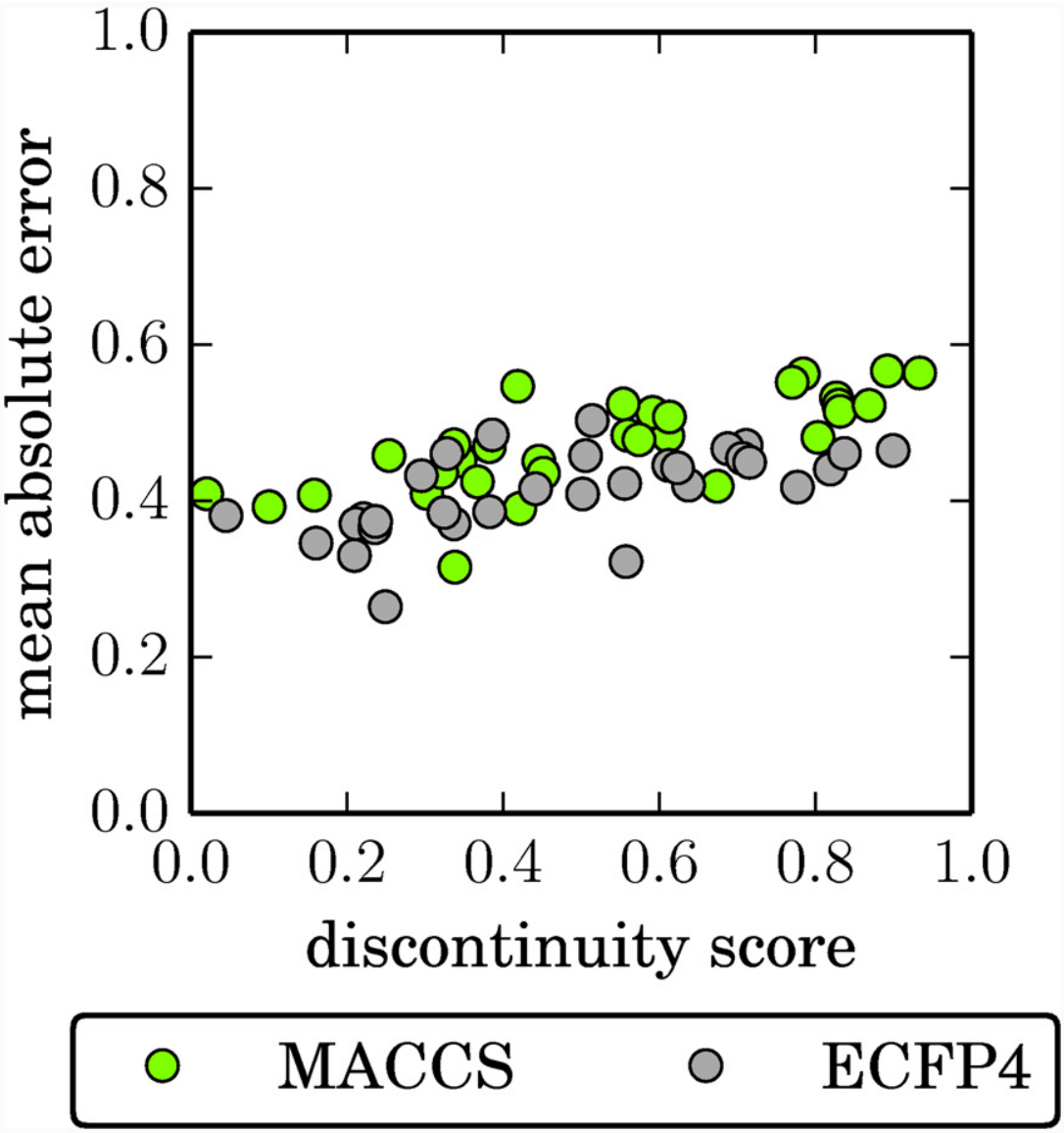


Fig 3. SAR discontinuity vs. SVR error. For each data set and fingerprint, the global discontinuity score is plotted against the mean absolute error of the SVR model, determined on the test set. Prediction errors display the tendency to increase with increasing discontinuity scores.

doi:10.1371/journal.pone.0119301.g003

to the respective scores calculated on the basis of experimental potency values. For score calculations on the basis of predicted values, the mean predicted potency of each test and training set compound was determined over all trials. The few compounds that never occurred in any of the randomly selected training or test sets were assigned their observed potency value.

Fig. 4 reports the comparison of predicted and observed SAR scores. Observed continuity scores of all data sets were almost perfectly reproduced (Fig. 4a). By contrast, predicted discontinuity scores were consistently and significantly lower than observed scores (Fig. 4b). For both MACCS and ECFP4, predicted test set discontinuity scores were very low (mostly below 0.1). The predicted training set scores were only slightly higher for MACCS, but substantially higher

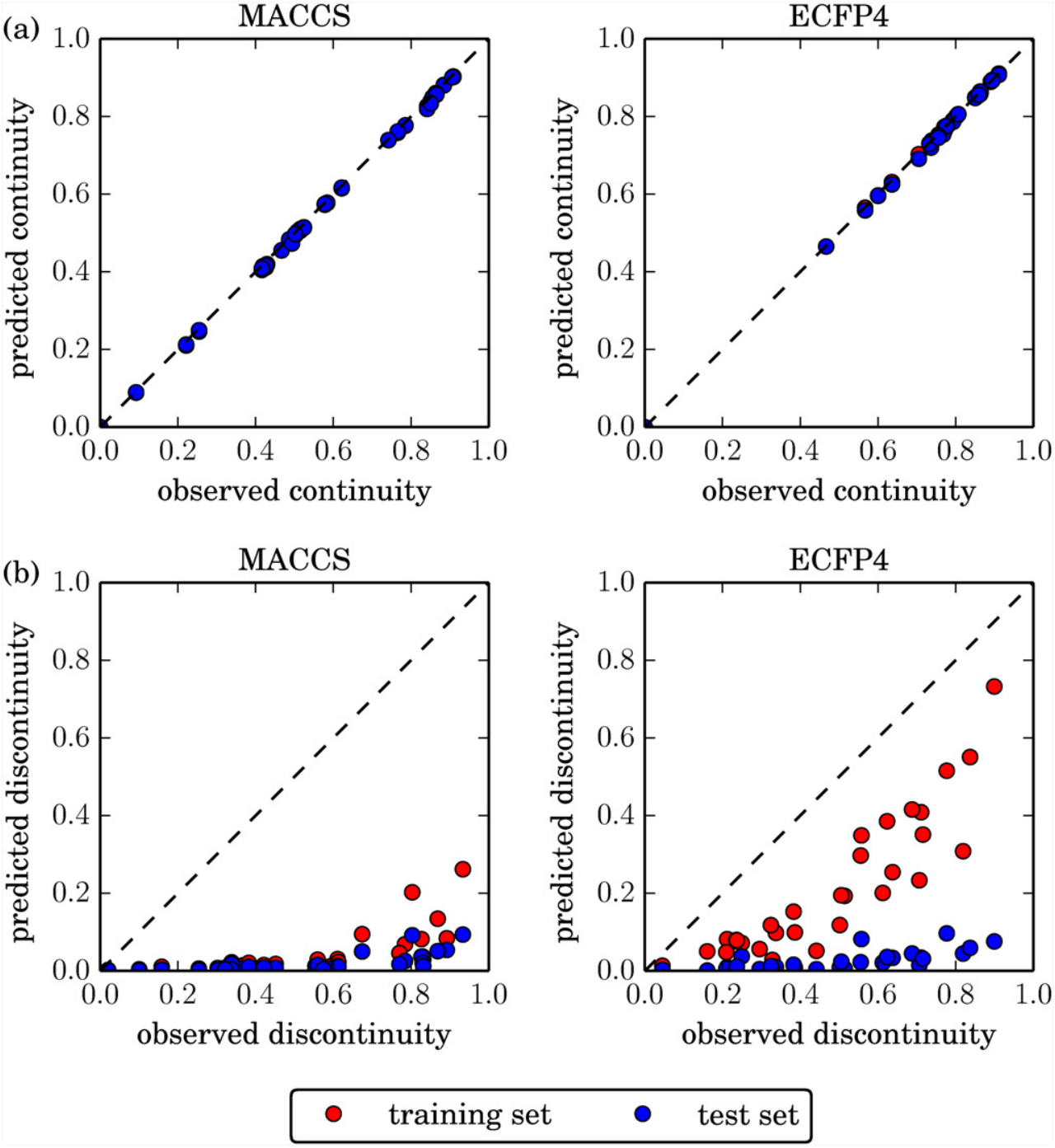


Fig 4. Observed vs. predicted SAR characteristics. The observed global (a) continuity and (b) discontinuity score of each data set and fingerprint is plotted against the predicted training set (red) and test set (blue) value (blue and red data points are placed in the fore- and background, respectively).

doi:10.1371/journal.pone.0119301.g004

for ECFP4. Nevertheless, even for ECFP4, scores for training set compounds were consistently underpredicted.

Limited generalization ability in discontinuous SAR regions. The comparison of predicted training and test set continuity scores showed that the SVR models fit continuous training data subsets very well and were generalizable for yielding high-quality potency prediction on continuous test data. In contrast, the comparison of predicted training and test set discontinuity scores indicated that only the ECFP4-based models partly fit discontinuous training data, but did not generalize well for potency predictions on test compounds in discontinuous SAR environments. Thus, SAR characteristics substantially influenced the quality of SVR-based predictions. Potency patterns in discontinuous SAR regions were generally difficult to predict.

Influence of regularization on discontinuity scores. In light of these findings, we also examined how the choice of the regularization term *C* influenced the magnitude of discontinuity scores based upon predicted potency values. Fig. 5 reports the discontinuity scores calculated on the basis of predicted test and training set potencies plotted against the regularization term for an exemplary data set (adenosine A3 receptor antagonists, TID 280). The horizontal black line denotes the observed discontinuity score calculated from the experimental potency values. Although observed discontinuity scores varied for all data sets, the trends in Fig. 5 were consistently detected. Discontinuity scores calculated with MACCS based upon predicted potencies gradually increased with increasing *C* values (consistent with the observation that training errors decreased with increasing *C*, as discussed above). In this case, the largest regularization terms yielded highest (albeit still substantially underpredicted) discontinuity scores. By contrast, for ECFP4, the discontinuity scores rapidly increased for small C values and then essentially remained constant, with much higher scores achieved on the basis of predicted training set than test set potency values.

Plausible rationale for potency prediction errors. Taking into consideration that the regularization term balances SVR model complexity with the amount of permitted training errors, we reasoned that the consistent underestimation of discontinuity scores might result from incorrectly predicted potency values in activity cliff regions, for the following reasons: Activity cliffs are generally rare in compound data sets involving on average only ~20% of active compounds [33]. Moreover, because activity cliffs consist of pairs of structurally analogous compounds with largest potency difference in a data set, only a small percentage of highly potent compounds (at most ~10%, but in practice often less [33]) is responsible for their formation. If training errors are permitted for these highly potent compounds, the SVR algorithm is expected to produce an easy to derive, low-complexity model yielding overall accurate predictions for many active compounds in low or intermediate potency ranges. This hypothesis is fully consistent with the regularization profile of ECFP4 reported in Fig. 5, which showed that a stable prediction model was obtained for small values of C, permitting training errors and a limited underfit of the model for the benefit of enabling low-complexity predictions.

SVR modeling effects on activity landscapes

In order to evaluate the hypothesis formulated above, we generated 3D activity landscapes [28] for our data sets on the basis of experimental potency values and compared them to corresponding landscapes generated on the basis of potency values predicted for the test sets. Since essentially all data set compounds with only few exceptions occurred multiple times (or at least once) in training and test sets (see Methods), these landscape views had conserved topology and could thus be directly compared. Fig. 6 shows ECFP4-based activity landscape comparisons for two exemplary data sets that illustrate consistently detected effects. These data sets and their observed landscapes were characterized by medium to high degrees of discontinuity

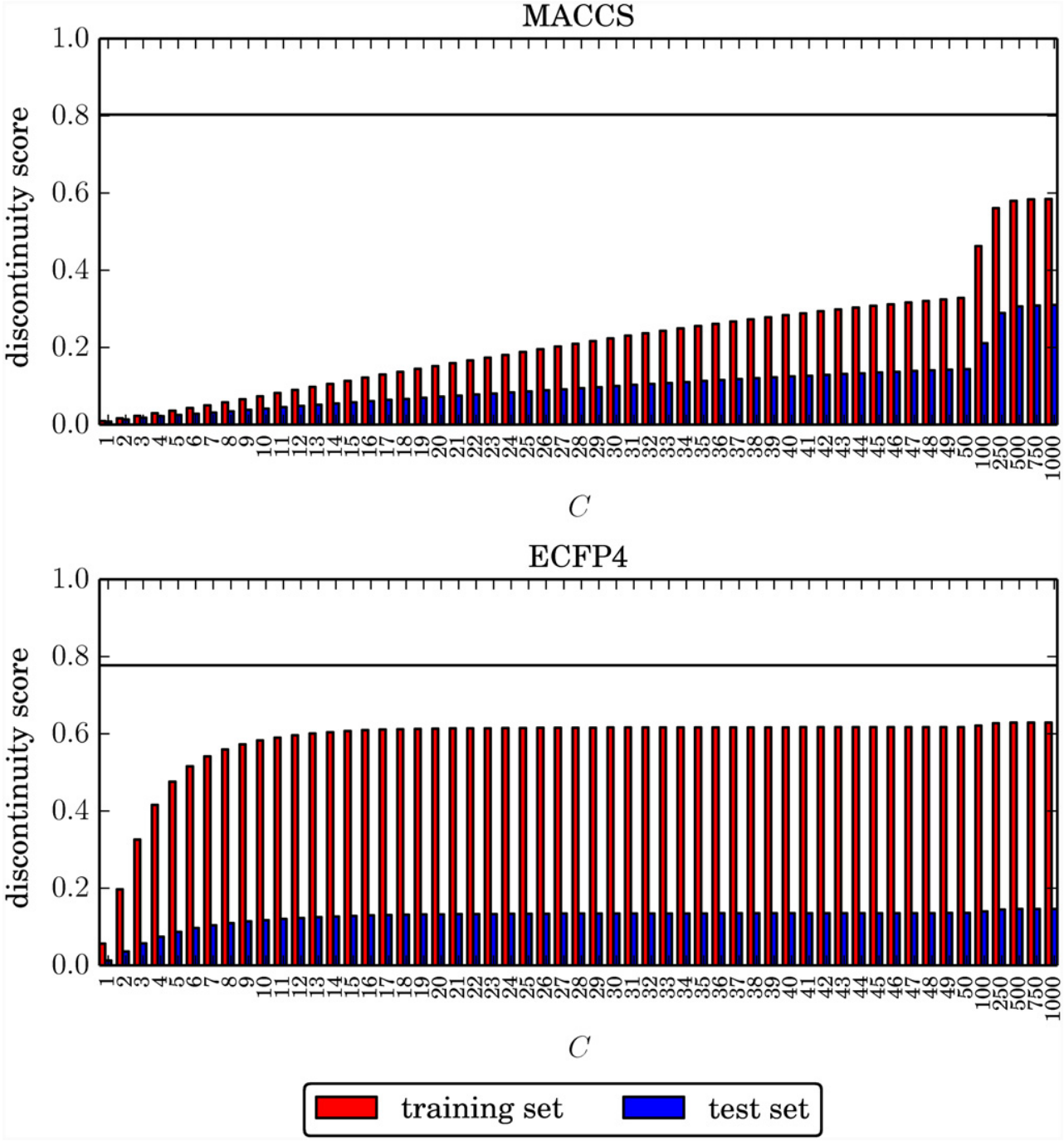


Fig 5. Exemplary discontinuity score profile. For each value of the regularization term *C*, the discontinuity score resulting from the predicted training and test set potency values is reported for data set TID 280. The black line denotes the observed discontinuity score of the data set. Regularization values are evenly spaced on the horizontal axis, regardless of their magnitude.

doi:10.1371/journal.pone.0119301.g005

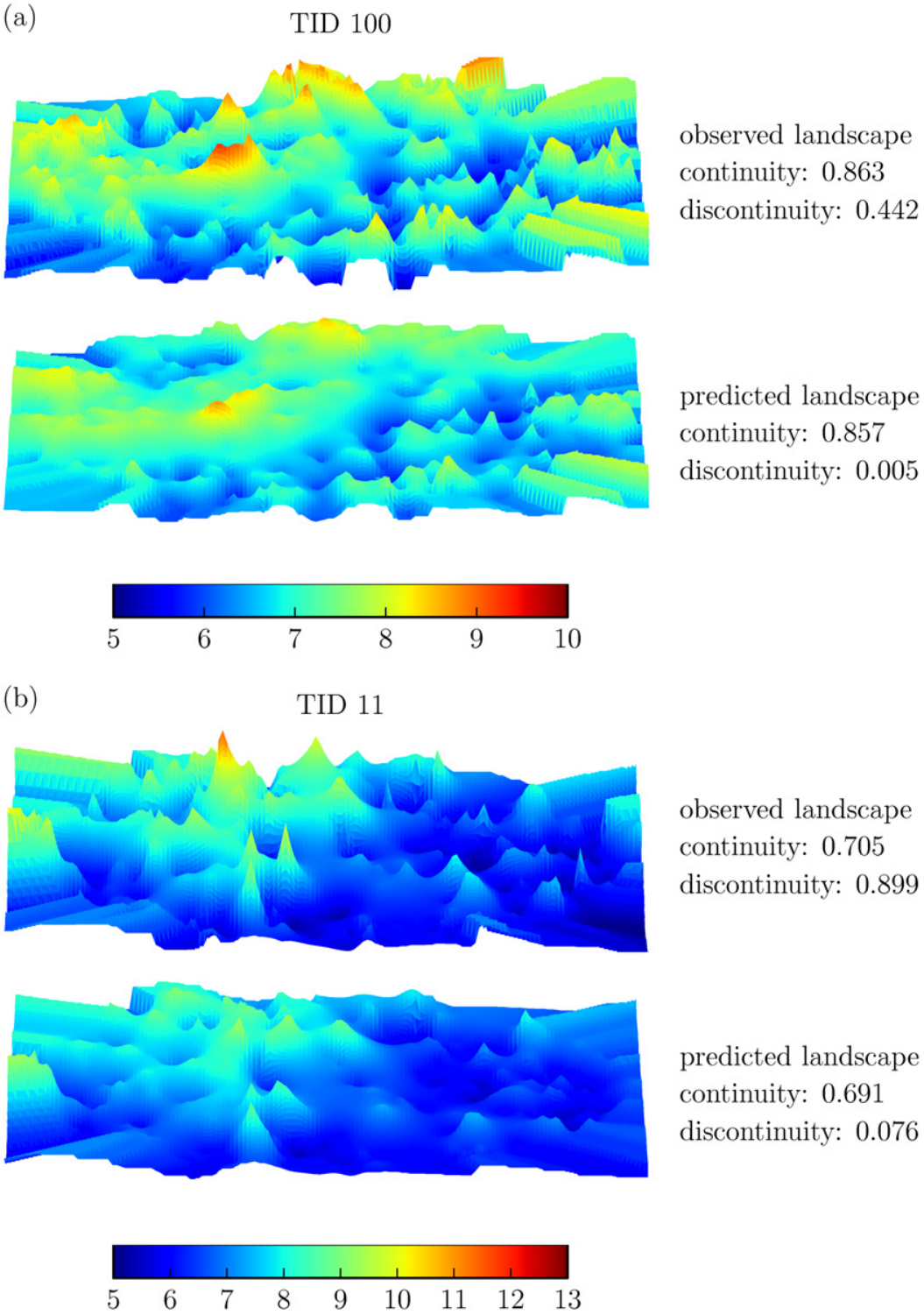


Fig 6. Observed and predicted activity landscapes. For two representative data sets, observed and predicted 3D activity landscape representations are compared and corresponding continuity and discontinuity scores are reported. Landscape surface elevation correlates with increasing potency. The activity landscape views are represented applying a continuous spectrum from blue to red spanning a potency range of (a) 5 to 10 pK_i (TID 100) and (b) 5 to 13 pK_i (TID 11). Hence, the positions of the most potent compounds in the landscapes are colored red.

doi:10.1371/journal.pone.0119301.g006

(resulting from pairwise potency relationship contributions of similar data set compounds). *Rugged regions* in the original landscapes delineate centers of SAR discontinuity where yellow-to-red *peaks* represent most potent data set compounds involved in the formation of activity cliffs with weakly potent structural analogs. In addition, blue *valleys* or blue-to-yellow *plateaus* in these landscapes delineate continuous SAR regions. Comparing the observed and predicted activity landscapes, major "smoothing" effects became apparent: In the predicted landscapes, the most prominent peaks in the original landscapes disappeared and rugged regions were flattened, consistent with the underprediction of SAR discontinuity. Artificial smoothing of activity landscapes directly resulted from incorrect predictions for the small proportion of highly potent compounds. For these compounds, much too low potency values were predicted by the SVR models and, consequently, the activity cliffs they formed were no longer detectable. Thus, consistent with conclusions drawn from SVR vs. SAR analysis, mispredictions of the most potent dataset compounds were the price to pay for obtaining statistically sound SVR models that yielded accurate predictions for the weakly to moderately potent data set compounds.

Conclusions

In this study, we have analyzed in detail the use of SVR models for compound potency prediction, which represents an increasingly popular QSAR strategy. A major attraction of the SVR approach and other kernel methods are their principal ability to account for non-linear SARs, which sets them apart from classical QSAR methods and enables potency predictions for structurally heterogeneous data sets on a large scale. In order to better understand intrinsic features, opportunities, and limitations of these SVR models, we have systematically analyzed general regression metrics (such as coefficients of determination and error values), model regularization, SAR characteristics, and observed vs. predicted activity landscapes. On the basis of our analysis, a detailed picture of SVR model performance has been obtained, providing a number of implications for practical applications.

For the wide spectrum of high-quality compound data sets with different SAR characteristics we investigated, SVR models with overall low prediction errors were obtained, without subjective intervention. For the majority of compounds across all data sets, potency values were correctly predicted within an order of magnitude, which is in accord with state-of-the-art QSAR standards and within the range of experimental assay variations. These findings provide general support for SVR modeling, consistent with a number of previous studies. We also found that SVR model performance was substantially influenced by the use of alternative molecular representations, which is a known conundrum of machine learning applications in chemoinformatics.

However, our detailed investigation of SVR model performance also yielded a number of new insights, pointing at critical issues that should merit careful consideration. For example, on the basis of regularization parameter analysis, SVR models were robust against data overfitting (consistently stable models were obtained for ECFP4), but generally vulnerable to underfitting. Furthermore, prediction errors were statistically correlated with increasing global SAR discontinuity of compound data sets. In light of these observations, we carefully examined SVR predictions and identified systematic errors in the prediction of highly potent compounds. Even with proper model regularization and a suitable molecular representation such as ECFP4, only highly potent *training set* compounds could be predicted approaching acceptable accuracy (cf. Fig. 5; only the discontinuity scores calculated on predicted training set potencies approach the true data set discontinuity). By contrast, the SVR models lacked the generalization ability to extrapolate from training data to new discontinuous test data and essentially failed to correctly predict high potency values for test compounds. These findings were rationalized by the

intrinsic feature of the SVR algorithm to strive for a balance between acceptable training error margins and model complexity. Because highly potent compounds typically represent only a small proportion of a data set, prediction errors can be algorithmically tolerated in these instances to derive a model that is of limited computational complexity but yields sufficiently accurate predictions for the majority of data set compounds. However, for QSAR applications, systematic errors in predicting highly potent compounds are a severe limitation, because such compounds represent prime prediction targets.

When comparing activity landscapes of data sets based upon experimental potency measurements with landscapes generated on the basis of SVR potency predictions, we found that prominent activity cliffs were eliminated by consistently predicting artificially low potency values for highly potent cliff compounds. Thus, SVR-based prediction of activity landscapes resulted in a substantial loss of SAR information. This also meant that the SVM paradigm of nonlinearity, albeit enabling meaningful predictions for structurally heterogeneous data sets as a whole, did not apply to modeling regions of high local SAR discontinuity formed by structural analogs with large potency variations, indicating a principal limitation of the approach.

In summary, our analysis has revealed that care must be taken when utilizing SVR for SAR/QSAR applications, despite promising SVR potency prediction statistics for many different compound data sets. For practical applications, the consistently incorrect prediction of highly potent compounds identified in our analysis represents a likely problem, because most attractive (highly potent) candidate compounds might be missed. However, these issues also present attractive opportunities for future method development, for example, the design of algorithmic SVR variants that would penalize prediction errors in most attractive property ranges.

Acknowledgments

The use of OpenEye's OEChem Toolkit was made possible by their free academic licensing program.

Author Contributions

Conceived and designed the experiments: J. Balfer J. Bajorath. Performed the experiments: J. Balfer. Analyzed the data: J. Balfer J. Bajorath. Wrote the paper: J. Balfer J. Bajorath.

References

1. Vapnik VN (2000) The Nature of Statistical Learning Theory, 2nd Ed. New York, Springer. 314p.
2. Byvatov E, Schneider G (2003) Support Vector Machine Applications in Bioinformatics. Applied Bioinformatics 2: 67–77. PMID: 15130823
3. Pavlidis P, Wapinski I, Noble WS (2004) Support Vector Machine Classification on the Web. Bioinformatics 20: 586–587. PMID: 14990457
4. Varnek A, Baskin I (2012) Machine Learning Methods for Property Prediction in Chemoinformatics: Quo Vadis? J Chem Inf Model 52: 1413–1437. doi: 10.1021/ci200409x PMID: 22582859
5. Vogt M, Bajorath J (2012) Chemoinformatics: A View of the Field and Current Trends in Method Development. Bioorg Med Chem 20: 5317–5323. doi: 10.1016/j.bmc.2012.03.030 PMID: 22483841
6. Drucker H, Burges C (1997) Support Vector Regression Machines. Adv Neural Inform Process Systems 9: 155–161.
7. Smola AJ, Schölkopf B (2004) A Tutorial on Support Vector Regression. Stat Comput 14: 199–222.
8. Cherkasov A, Muratov EN, Fourches D, Varnek A, Baskin I, Cronin M, et al. (2014) QSAR Modeling: Where Have You Been? Where Are You Going To? J Med Chem 57: 4977–5010. doi: 10.1021/jm4004285 PMID: 24351051
9. Xue CX, Zhang RS, Liu HX, Yao XJ, Liu MC, Hu ZD, et al. (2004) QSAR Models for the Prediction of Binding Affinities to Human Serum Albumin Using the Heuristic Method and a Support Vector Machine. J Chem Inf Comput Sci 44: 1693–1700. PMID: 15446828

10. Yuan Y, Zhang R, Hu R, Ruan X (2009) Prediction of CCR5 Receptor Binding Affinity of Substituted 1-(3,3-diphenylpropyl)-piperidinyl Amides and Ureas Based on the Heuristic Method, Support Vector Machine and Projection Pursuit Regression. Eur J Med Chem 44: 25–34. doi: 10.1016/j.ejmech.2008.03.004 PMID: 18433938
11. Sun M, Chen J, Wei H, Yin S, Yang Y, Ji M (2009) Quantitative Structure-activity Relationship and Classification Analysis of Diaryl Ureas Against Vascular Endothelial Growth Factor Receptor-2 Kinase Using Linear and Non-linear Models. Chem Biol Drug Des 73: 644–654. doi: 10.1111/j.1747-0285.2009.00814.x PMID: 19635056
12. Lind P, Maltseva T (2003) Support Vector Machines for the Estimation of Aqueous Solubility. J Chem Inf Comput Sci 43: 1855–1859. PMID: 14632433
13. Song M, Clark M (2005) Development and Evaluation of an in Silico Model for hERG Binding. J Chem Inf Model 46: 392–400.
14. Fatemi MH, Gharaghani S (2007) A Novel QSAR Model for Prediction of Apoptosis-inducing Activity of 4-aryl-4-H-chromenes Based on Support Vector Machine. Bioorg Med Chem 15: 7746–7754. PMID: 17870538
15. Leong MK (2007) A Novel Approach Using Pharmacophore Ensemble/Support Vector Machine (PhE/SVM) for Prediction of hERG Liability. Chem Res Toxicol 20: 217–226. PMID: 17261034
16. Gombar VK, Hall SD (2013) Quantitative Structure-activity Relationship Models of Clinical Pharmacokinetics: Clearance and Volume of Distribution. J Chem Inf Model 53: 948–957. doi: 10.1021/ci400001u PMID: 23451981
17. Wassermann AM, Wawer M, Bajorath J (2010) Activity Landscape Representations for Structure-Activity Relationship Analysis. J Med Chem 53: 8209–8223. doi: 10.1021/jm100933w PMID: 20845971
18. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, et al. (2012) ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Res 40: D1100–D1107. doi: 10.1093/nar/gkr777 PMID: 21948594
19. Baell JB, Holloway GA (2010) New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. J Med Chem 53: 2719–2740. doi: 10.1021/jm901137j PMID: 20131845
20. Balfer J, Bajorath J (2015) 31 ChEMBL data sets for regression modeling. ZENODO. http://dx.doi.org/10.5281/zenodo.13986
21. MACCS Structural keys (2011) Accelrys, San Diego, CA.
22. Rogers D, Hahn M (2010) Extended-connectivity Fingerprints. J Chem Inf Model 50: 742–754. doi: 10.1021/ci100050t PMID: 20426451
23. OEChem TK version 2.0.0. OpenEye Scientific Software, Santa Fe, NM. http://www.eyesopen.com.
24. Peltason L, Bajorath J (2007) SAR Index: Quantifying the Nature of Structure-Activity Relationships. J Med Chem 50: 5571–5578. PMID: 17902636
25. Rogers DJ, Tanimoto TT (1960) A Computer Program for Classifying Plants. Science 132: 1115–1118. PMID: 17790723
26. Willett P, Barnard J, Downs GM (1998) Chemical Similarity Searching. J Chem Inf Comput Sci 38: 983–996.
27. Dimova D, Stumpfe D, Bajorath J (2013) Quantifying the Fingerprint Descriptor Dependence of Structure-Activity Relationship Information on a Large Scale. J Chem Inf Model 53: 2275–2281. doi: 10.1021/ci4004078 PMID: 23968259
28. Peltason L, Iyer P, Bajorath J (2010) Rationalizing Three-dimensional Activity Landscapes and the Influence of Molecular Representations on Landscape Topology and Formation of Activity Cliffs. J Chem Inf Model 50: 1021–1033. doi: 10.1021/ci100091e PMID: 20443603
29. Cortes C, Vapnik VN (1995) Support Vector Networks. Machine Learning 20: 273–297.
30. Boser BE, Guyon IM, Vapnik VN (1992) A Training Algorithm for Optimal Margin Classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory: Pittsburgh, Pennsylvania. pp. 144–152.
31. Ralaivola L, Swamidass SJ, Saigo H, Baldi P (2005) Graph Kernels for Chemical Informatics. Neural Netw 18: 1093–1110. PMID: 16157471
32. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. (2011) Scikit-learn: Machine Learning in Python. J Mach Learn Res 12: 2825–2830.
33. Stumpfe D, Hu Y, Dimova D, Bajorath J (2014) Recent Progress in Understanding Activity Cliffs and Their Utility in Medicinal Chemistry. J Med Chem 57: 18–28. doi: 10.1021/jm401120g PMID: 23981118

Summary

In this study, we have investigated the quantitative and qualitative performance of SVR models for compound potency prediction. While their global performance in terms of R^2 scores was acceptable, we found several limiting factors for the quality of the models.

First, our analysis revealed a correlation between mean absolute error of the models and discontinuity score of the data sets. It seems to be intuitive that compound data sets with higher discontinuity scores are harder to model by computational means; however, kernelized methods are often portrayed as able to deal with any kind of nonlinearity or discontinuity, which they are obviously not. The dependency of potency modeling success on SAR discontinuity has to our best knowledge not been shown before.

Second, we showed that while global data set continuity of varying magnitude could be well preserved by regression modeling, discontinuity was constantly underpredicted. This effect cannot simply be attributed to a poor regularization choice, as our analysis clearly showed. It rather originated from the fact that high discontinuity is caused by few highly potent compounds in the chemical neighborhood of many intermediate or weakly potent ones. As a global optimization algorithm, SVR seeks to minimize the overall training error, leading to a constant underprediction of highly potent ligands. While this can be the desirable behavior in a number of application scenarios, it clearly is not for potency prediction in the lead optimization stage. Here, one is interested most in the few highly potent compounds and their neighboring SAR regions, as they serve as focal points for SAR analysis.

In summary, our findings implicate that care must be taken when SVR models are applied for the prediction of compound potency. Beyond global statistics, it is important to build models that generalize well on interesting local regions. This could be achieved by modifying optimization error functions or building discontinuity-sensitive kernels. We hope that our study can contribute to a higher awareness of the problem of mispredictions in underrepresented, yet important, SAR regions.

As this chapter has shown, not only the overall performance of an LBVS model is of interest. It is also important to understand the algorithmic procedures to be able to identify and circumvent possible mispredictions. Hence, the next part of this thesis deals with the interpretation of prediction models. In the upcoming chapters, two intuitive methods for the analysis and interpretation of naïve Bayes and SVM models are introduced.

Part III

Interpretation of Predictors for Virtual Screening

5

Introduction of a Methodology for Visualization and Graphical Interpretation of Bayesian Classification Models

Introduction

In the last chapter, we have used 3D activity landscapes to visualize predictions and qualitatively compare observed and predicted SAR characteristics. For SAR analysis, many visualization methods have been developed to provide an intuitive assessment for data analysis and augment quantitative methods. However, not much effort has been put into an analogously intuitive assessment of machine learning models. The next study addresses this issue by introducing a visualization method for naïve Bayes classification models. As such, it presents both models and individual predictions made using naïve Bayes classifiers in a directly accessible way. The abstract model and prediction visualizations aim to bridge the gap between statistical analysis, which is carried out by the machine learning expert, and the ligand-centric view of the medicinal chemist. While statistical analysis is used to create the interactive visualization, the prediction visualization can be transferred into a weight mapping onto the molecular graph.

Reprinted with permission from

Balfer, J.; Bajorath, J. Introduction of a Methodology for Visualization and Graphical Interpretation of Bayesian Classification Models. *J. Chem. Inf. Model.* **2014**, *54*, 2451–2468.

JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

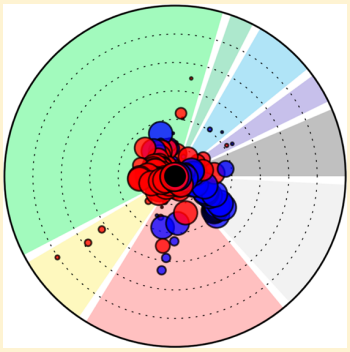
pubs.acs.org/jcim

Introduction of a Methodology for Visualization and Graphical Interpretation of Bayesian Classification Models

Jenny Balfer and Jürgen Bajorath*

Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Dahlmannstrasse 2, D-53113 Bonn, Germany

ABSTRACT: Supervised machine learning models are widely used in chemoinformatics, especially for the prediction of new active compounds or targets of known actives. Bayesian classification methods are among the most popular machine learning approaches for the prediction of activity from chemical structure. Much work has focused on predicting structure−activity relationships (SARs) on the basis of experimental training data. By contrast, only a few efforts have thus far been made to rationalize the performance of Bayesian or other supervised machine learning models and better understand why they might succeed or fail. In this study, we introduce an intuitive approach for the visualization and graphical interpretation of naïve Bayesian classification models. Parameters derived during supervised learning are visualized and interactively analyzed to gain insights into model performance and identify features that determine predictions. The methodology is introduced in detail and applied to assess Bayesian modeling efforts and predictions on compound data sets of varying structural complexity. Different classification models and features determining their performance are characterized in detail. A prototypic implementation of the approach is provided.



INTRODUCTION

Machine learning models are used for a variety of applications in chemoinformatics including, for instance, the prediction of compound activity and other molecular properties or biological targets of known actives.[1−3] When a set of known positive (active) and negative (inactive) training compounds is available, supervised machine learning is an approach of choice for building predictive models of activity.[1,2] In the chemoinformatics community, the currently most frequently applied supervised machine learning methods include random forests, support vector machines, and Bayesian classifiers.[1,2] Random forest models utilize ensembles of decision trees to arrive at consensus predictions, support vector machines derive separating hyperplanes for class label prediction in feature spaces of increasing dimensionality, and Bayesian classifiers are probabilistic models based upon Bayes theorem.

In general, supervised learning is used to build complex models from training data that go beyond the derivation of simple rules to determine a classification outcome. Model building requires the definition of a suitable molecular descriptor space for classification and the selection of preferred models on the basis of preset performance criteria. If sufficient training data is available and meaningful reference (feature) spaces can be generated, effective models can often be derived for a variety of classification or regression tasks.[4−7]

Areas in which supervised learning is applied can roughly be divided into those in which a computer should learn a concept that is intuitively known to humans as opposed to those where the concept itself is not fully understood by human experts. An example for the first area is image classification. Humans can usually identify and distinguish different objects in images. Yet, successful recognition is the result of very complex reasoning and neural functions, which cannot be easily transferred to a computer.[8] Hence, supervised classification is applied in such situations to let the computer "learn" the concept from examples. However, many chemoinformatics problems fall into the second area mentioned above. For example, even an experienced medicinal chemist can typically not predict the activity of given compounds against biological targets in a consistent manner and without error. Thus, much research is dedicated to rationalizing and predicting structure−activity relationships (SAR)[9−11] as there are no generally applicable rules governing compound-target interactions that could be consistently applied. Even in the presence of significant amounts of experimental data, the exact mechanism of compound-target interactions is often difficult to determine.[12] Therefore, if a machine learning model for activity prediction can be derived, it should be important to understand the characteristics of the model that determine its decisions. However, this is in general difficult to accomplish. Clearly, obtaining such insights would help to reduce or eliminate the well-known "black box" character of many machine learning approaches, which often limits their utility. In interdisciplinary research, gaining insights into the mechanisms by which computer models function is often a prerequisite of their acceptance and for the willingness to build experimental projects around predictions. Hence, the importance of chemical interpretability of machine learning models and their predictions should not be underestimated.

Received: July 9, 2014
Published: August 19, 2014

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

Molecular feature spaces used in chemoinformatics are typically large and high-dimensional, as millions of biologically relevant compounds and thousands of chemical descriptors are available.[13] For the navigation of such feature spaces and property prediction, naïve Bayesian classifiers are often applied.[14−25] Their popularity can be attributed to their relatively simplistic design, the ability to efficiently operate on large and high-dimensional data sets, and their limited sensitivity to data noise; an important aspect for chemoinformatics applications.[14,15] In recent years, naïve Bayesian classifiers have been applied to identify therapeutically relevant targets[16] as well as novel active compounds for given targets,[17−20] further improve docking scores,[15,21−23] or predict absorption, distribution, metabolism, and excretion (ADME) properties[24] and multidrug resistance reversal activity.[25] For such studies, compounds have mostly been represented using binary fingerprints.[14−20,26]

Although a number of successful naïve Bayesian models have been reported, only very few studies have thus far attempted to address the question how exactly these models work and why they might succeed or fail. As Klon et al. point out in their study to predict ADME properties, "understanding why a compound has undesirable ADME characteristics is just as important as knowing that it does".[24] These authors have also aimed to rationalize their classification models. For instance, it was attempted to explain the success of a naïve Bayesian classifier in enriching favorable docking scores by training alternative models on only subsets of features from preferred models.[21] Other investigators have addressed the interpretability issue by using intuitive molecular representations such as chemical fragment descriptors[19] or by focusing on specific compounds whose activity could only be predicted using naïve Bayesian classification or other machine learning approaches.[27]

In this study, we introduce an intuitive approach for the visualization and graphical interpretation of naïve Bayesian classification models. Previous work on graphical interpretation of machine learning models has primarily focused on depicting features in heat maps[28] or similarity maps[29] that are important for prediction of individual molecules. The methodology introduced herein also enables the assessment of individual predictions but goes far beyond the analysis of single compounds by providing a visualization scheme for an entire classification model. Furthermore, it also reveals the contributions of features that are absent in test compounds.

CONCEPTS AND METHODS

Naïve Bayesian classification. The naïve Bayesian classifier makes use of Bayes' theorem to predict the probability $P(y|x)$ of an instance x to belong to class y:[30]

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{1}$$

It is called naïve because it assumes all features x_d in x to be independent of each other;[31] applying this feature independence assumption, eq 1 can be rewritten as

$$P(y|x) = \frac{\prod_d P(x_d|y)P(y)}{\prod_d P(x_d)} \tag{2}$$

To build a naïve Bayesian model, a training set of labeled instances with different class labels is utilized for supervised learning. Although there are no principal assumptions concerning the nature of x and y, one often focuses on binary features and class labels, i.e., $x,y \in \{0,1\}$, for example, "active" vs "inactive". In this case, given a set of n training instances X and corresponding labels Y, the terms required for naïve Bayesian classification can be estimated as follows:[30]

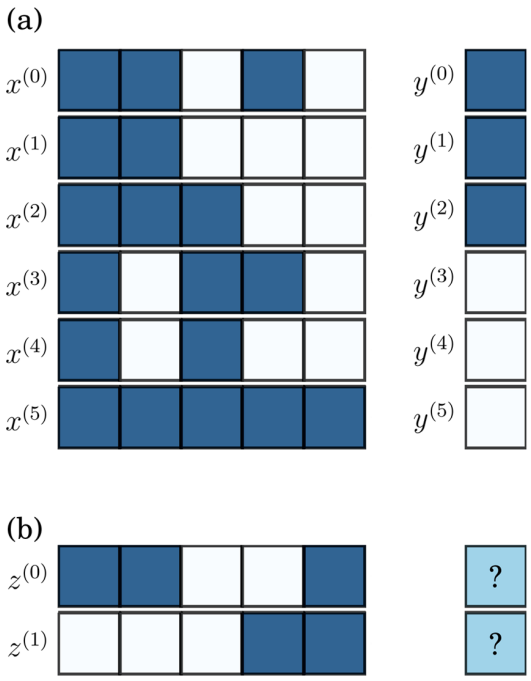


Figure 1. Motivating example. Shown is a theoretical "minimalist" example of a training and test set for supervised classification, represented as binary features. Blue squares indicate that a feature is set and white squares that it is not set. (a) Six training examples are given; three of which are positive, and the remaining three are negative. (b) The test set contains two examples with unknown class labels.

Table 1. Estimated Parameters for a Naïve Bayesian Model[a]

d	$P(x_d = 1\|y = 0)$	$P(x_d = 1\|y = 1)$	$P(y = 1)$
0	0.9688	0.9688	
1	0.3438	0.9688	
2	0.9688	0.3438	0.5
3	0.6562	0.3438	
4	0.3438	0.0313	

[a]Reported are the conditional feature probabilities and the prior probability for the model of the motivating example discussed in the text.

$$P(x_d = 1|y = \hat{y}) = \frac{\sum_i x_d^{(i)} \delta_{y^{(i)}\hat{y}} + \alpha}{\sum_i \delta_{y^{(i)}\hat{y}} + 2\alpha} \tag{3}$$

$$P(y = 1) = \frac{\sum_i y^{(i)}}{n} \tag{4}$$

Here, δ_{ij} is the Kronecker delta function, which is 1 for $i = j$ and 0 otherwise. The notations $x^{(i)}$ and $y^{(i)}$ refer to the i'th training instance and label, respectively. The term α is a Laplacian smoothing factor used to prevent the introduction of ill-defined probabilities, e.g., if a feature is never set in the training data. Since both class labels y and features x_d are binary, we can infer

$$P(x_d = 0|y = \hat{y}) = 1 - P(x_d = 1|y = \hat{y}) \tag{5}$$

$$P(y = 0) = 1 - P(y = 1) \tag{6}$$

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

Table 2. Odds Ratios for a Naïve Bayesian Model[a]

d	OR_d	$\log OR_d$
0	1.00	0.00
1	2.82	1.04
2	0.35	−1.04
3	0.52	−0.65
4	0.09	−2.40

[a]Reported are the odds ratios of features x_0−x_4 in the model of the motivating example. The closer the odds ratio of a given feature is to 1, the smaller is its influence on the classification.

The meaning of these equations can be illustrated by a simple example. Let us derive a naïve Bayesian model from a theoretical training set of three positive and negative examples each, which are each represented by five features, as illustrated in Figure 1a. In the following, we refer to this example as the "motivating example" because it can be used to illustrate the opportunities of model visualization.

By applying eqs 3 and 4 and by setting $\alpha = 0.1$, the probabilities reported in Table 1 are obtained. Using eqs 5 and 6, all missing probabilities are derived. One now can make predictions for test

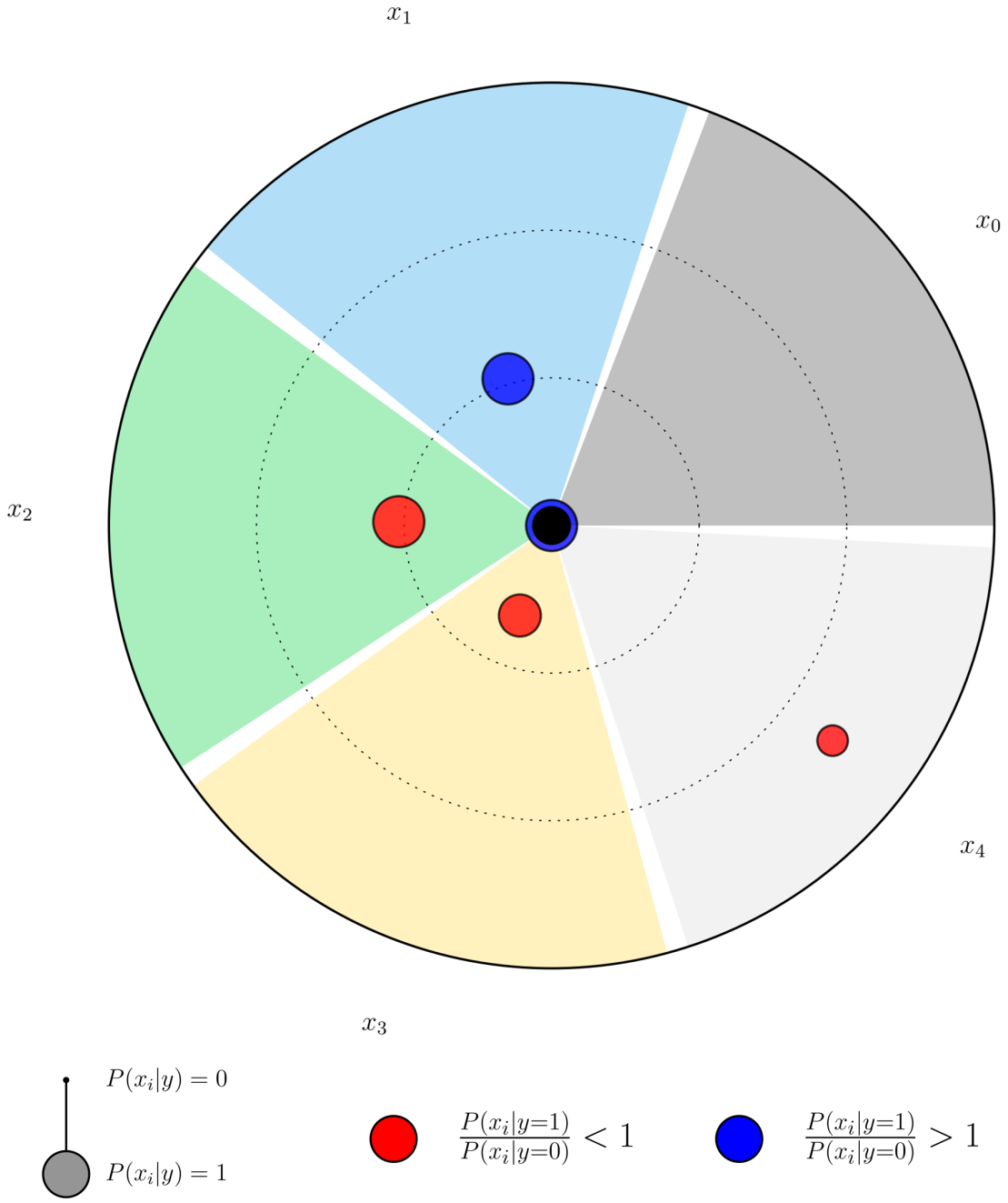


Figure 2. Principles of model visualization. The model for the motivating example is visualized. Each point represents a feature, and the distance to the pole corresponds to its absolute log odds ratio. Positive and negative influence on the classification is indicated by color coding and likelihood by size scaling, as detailed in the text.

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

instances such as $z^{(0)}$ and $z^{(1)}$ in Figure 1b. For each instance, we can calculate the class likelihood $P(z^{(i)}|y = \hat{y})$ and the evidence $P(z^{(i)})$ as follows:

$$P(z^{(0)}|y = 0) = \prod_d P(x_d = z_d^{(0)}|y = 0) = 0.00123$$

$$P(z^{(0)}|y = 1) = \prod_d P(x_d = z_d^{(0)}|y = 1) = 0.01263$$

$$P(z^{(1)}|y = 0) = \prod_d P(x_d = z_d^{(1)}|y = 0) = 0.00014$$

$$P(z^{(1)}|y = 1) = \prod_d P(x_d = z_d^{(1)}|y = 1) = 0.00001$$

$$P(z^{(0)}) = \sum_{\hat{y}} P(\hat{y}) \prod_d P(x_d = z_d^{(0)}|\hat{y}) = 0.0069$$

$$P(z^{(1)}) = \sum_{\hat{y}} P(\hat{y}) \prod_d P(x_d = z_d^{(1)}|\hat{y}) = 0.0001$$

The posterior probabilities are then given as

$$P(y = 0|z^{(0)}) = \frac{P(z^{(0)}|y = 0)P(y = 0)}{P(z^{(0)})} = 0.0887$$

$$P(y = 1|z^{(0)}) = \frac{P(z^{(0)}|y = 1)P(y = 1)}{P(z^{(0)})} = 0.9113$$

$$P(y = 0|z^{(1)}) = \frac{P(z^{(1)}|y = 0)P(y = 0)}{P(z^{(1)})} = 0.9545$$

$$P(y = 1|z^{(1)}) = \frac{P(z^{(1)}|y = 1)P(y = 1)}{P(z^{(1)})} = 0.0455$$

In this case, the positive class label would be predicted for the first and the negative class label for the second test instance.

Model Interpretation. The motivating example discussed above illustrates two important points: First, for classification, the marginal probability $P(z)$ is only used as a normalization factor and can hence be omitted if knowledge of exact posterior probabilities is not required. In fact, it has been shown that successful naïve Bayesian classification models often produce rather poor probability estimates.[32] Given that exact probability values are not required, the classification rule can be simplified:

$$y = \arg\max_{\hat{y} \in Y} P(x|y = \hat{y})P(\hat{y}) \tag{7}$$

Second, prior class probabilities, which might be utilized to incorporate user knowledge or a measure of data imbalance, are relatively easy to interpret. However, estimated class likelihoods mostly determine the classification decision. Each of the c class likelihoods are a product of d conditional feature probabilities, with c being the number of classes and d the number of dimensions. Unfortunately, these probabilities cannot be easily interpreted. This is the case because a high conditional feature probability does not necessarily indicate that a given feature is important for predicting a certain class and a low probability does not always mean that the feature is irrelevant. For example, let us consider feature x_0 of the motivating example. It is set in all instances, regardless of the class label, and thus has a high conditional feature probability for both the positive and the negative class. This feature provides no relevant information for classification, and the same applies to features that are never (or almost never) set in the training data, such as feature x_4. On the other hand, features x_1 and x_2 are always set in one of the classes and only once in the respective other class. In this case, the conditional feature probabilities of the active or inactive class, respectively, are three times higher than of the other one, which renders these two features highly descriptive.

A formal way to account for these feature probability relationships is provided by the so-called "odds ratio" (OR) of conditional probabilities:

$$\mathrm{OR}_d = \frac{P(x_d = 1|y = 1)}{P(x_d = 1|y = 0)} \tag{8}$$

The odds ratios for the motivating example are reported in Table 2. The approach of considering the odds ratios is theoretically established by rearranging the classification rule:

$$y = \arg\max_{\hat{y} \in Y} \prod_d P(x_d|y = \hat{y})P(\hat{y})$$

$$\Leftrightarrow y = \begin{cases} 1 & \text{if } \prod_d P(x_d|y = 1)P(y = 1) > \prod_d P(x_d|y = 0)P(y = 0) \\ 0 & \text{otherwise} \end{cases}$$

$$\Leftrightarrow y = \begin{cases} 1 & \text{if } \prod_d \frac{P(x_d|y = 1)}{P(x_d|y = 0)} > \frac{P(y = 0)}{P(y = 1)} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

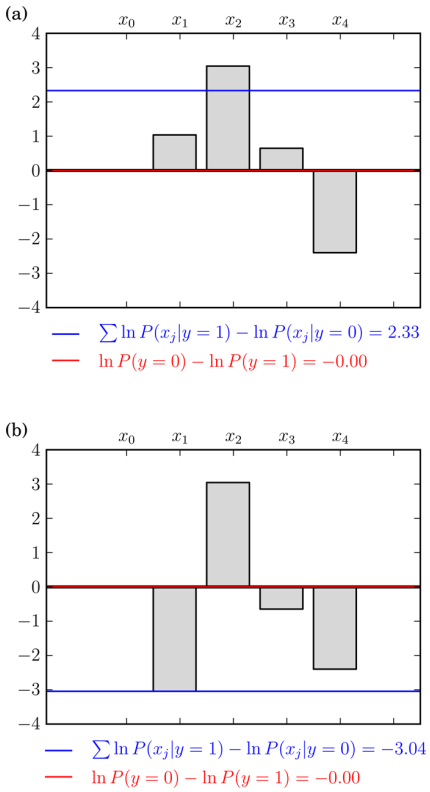


Figure 3. Principles of prediction visualization. The prediction for test instances (a) $z^{(0)}$ and (b) $z^{(1)}$ is visualized. Each bar represents the log odds ratio of a given feature for the test instance. Their sum is reported as a blue line and the difference between logarithmic prior probabilities as a red line.

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

The larger the odds ratio is, the higher is a feature's influence on the class likelihood; on the other hand, the smaller the odds ratios are, the smaller the class likelihood becomes. The closer the ratio is to 1, the less classification information is associated with the feature. Because two features with $OR_i = x$ and $OR_j = 1/x$ encode the same magnitude of classification information, a log transformation is applied such that $\log OR_i = -\log OR_j$ (cf. Table 2; in log space, features x_1 and x_2 have equal influence). The absolute value of the log odds ratio can then be regarded as a measure of an individual feature's importance for a given classification task. If it is negative, the presence of the feature is an indicator of the negative class label; if it is positive, it is an indicator of the positive class label.

Visualization. As rationalized in the previous section, a naïve Bayesian model contains only two parameters that must be learned for classification: the *class prior probabilities* (priors) and the *class likelihood*. For binary classification, there are two class priors that are related as follows:

$$P(y = 0) = 1 - P(y = 1) \tag{10}$$

However, there are four conditional probabilities for each dimension (feature) of the input space. For each pair of probabilities, eq 5 applies: $P(x_d = 0|y = \hat{y}) = 1 - P(x_d = 1|y = \hat{y})$.

If one would like to understand how a given model reaches a decision, one has to consider the odds ratio of each dimension in the input space.

Our primary goal is the visualization and interactive graphical analysis of a naïve Bayesian classification model with Bernoulli features. For *model visualization*, we introduce a scatter plot of its input dimensions using polar coordinates. The area of the plot is subdivided by features. Each point $p = (r,\theta)$ represents one dimension of the input space. Its radius is determined by the absolute value of its log odds ratio in the model, and the angles of all points are evenly distributed over the interval $[0,2\pi]$. Hence, the larger the distance between a point and the pole, the more important it is for classification. Coloring distinguishes features that indicate the positive class from features indicating the negative class, i.e., features with a negative log odds ratio are colored red and features with positive log odds ratio blue. Furthermore, points in the plot are scaled in size according to their maximum conditional probability for one of the classes:

$$s = \max\{P(x_d = 1|y = 0), P(x_d = 1|y = 1)\} \tag{11}$$

Therefore, it is possible to distinguish features occurring in most of the examples in one class from those occurring only in a few examples. The log odds ratio alone does not account for these different frequencies of occurrence.

In Figure 2, the model for our motivating example is visualized. It is evident that feature x_4 mostly determines the prediction, whereas features x_1–x_3 are less important and feature x_0, which maps to the pole, is not relevant. Furthermore, features x_2–x_4 (red) support negative class label predictions, whereas x_1 (blue) supports positive class label predictions (given its positive log odds ratio).

In addition to global model visualization, it is also possible to visualize and interpret individual predictions, which is relevant for assessing unexpected predictions and for model refinement.

For *prediction visualization*, one can exploit the fact that an instance obtains a positive class label if the following inequality applies (cf. eq 9):

$$\prod_d \frac{P(x_d|y = 1)}{P(x_d|y = 0)} > \frac{P(y = 0)}{P(y = 1)} \tag{12}$$

This inequality can also be expressed in log space:

$$\begin{aligned} &\log\left(\prod_d \frac{P(x_d|y = 1)}{P(x_d|y = 0)} > \frac{P(y = 0)}{P(y = 1)}\right) \\ &= \sum_d \log P(x_d|y = 1) - \log P(x_d|y = 0) \\ &> \log P(y = 0) - \log P(y = 1) \end{aligned} \tag{13}$$

Through the log transformation the product in eq 12 becomes a sum. Accordingly, a single prediction can be represented in a bar chart. In this chart, each bar represents a given feature, and the difference between the conditional probability given the active and inactive class is plotted. For *model visualization*, the conditional probability $P(x_d = 1|y = \hat{y})$ is utilized, as discussed above. However, for *prediction visualization*, we use the actual probability $P(x_d = z_d|y = \hat{y})$, with z being the example to be predicted. In addition, the sum of log probabilities is reported by a blue line and the sum of log priors by a red line. Hence, the final classification decision can be visualized in combination with the features that mostly influence the decision.

Figure 3 shows the prediction visualization for our motivating example. The priors for both classes are constant, but the class likelihood changes as a consequence of different input data. The test instances $z^{(0)}$ and $z^{(1)}$ differ in dimensions x_0, x_1, and x_3. Because x_0 has no impact on the classification (which can also be inferred from model visualization in Figure 2), it is not shown in Figure 3. For both instances, the fact that x_2 is not set in the training examples (Figure 1b) serves as an indicator for the positive class (i.e., it results in a positive odds ratio of x_2 in Figure 3), whereas the presence of x_4 is indicative of the negative class (i.e., it results in a negative odds ratio of x_4). Furthermore, features x_1 and x_3 in $z^{(0)}$ make small contributions to the overall class likelihood. Taken together, these probabilities result in a positive class label prediction for $z^{(0)}$. By contrast, in $z^{(1)}$, x_1 is not set and x_3 is set, which results in a large negative contribution to the likelihood for x_1 and a smaller negative contribution for x_3. As a consequence, the sum of log likelihoods falls below the sum of log priors. Accordingly, for $z^{(1)}$, the negative class label is predicted.

MATERIALS AND PROTOCOLS

Compound Data Sets. Three compound data sets of increasing complexity were used for Bayesian modeling and visualization. These data sets included two sets from ChEMBL (version 18),[33] i.e., carbonic anhydrase I inhibitors (CAI) and calcitonin gene-related peptide type 1 receptor ligands (CGRPR), and, in addition, a set of ATP-site directed inhibitors primarily focused on mitogen-activated protein kinase 14

Table 3. Compound Data Sets[a]

data set	no. of compds	no. of BMS	no. of CSK
CAI	1306	407	179
MAPK14 (active)	265	86	45
MAPK14 (inactive)	164	80	45
CGRPR	305	133	78

[a]For all three data sets, the number of compounds, unique Bemis-Murcko scaffolds (BMS), and corresponding carbon skeletons (CSK) is reported. For MAPK14, active and confirmed inactive compounds are listed separately. The other two data sets only consist of active compounds. In these cases, a random subset of ChEMBL was used as inactive compounds (see text). Scaffolds were calculated using OpenEye's OEChem toolkit.[40]

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

Table 4. Model Performance[a]

	no. of training compds		no. of test compds				
data set	active	inactive	active	inactive	precision	recall	F1-score
CAI	1044	8000	262	2000	0.5360	0.9084	0.6742
MAPK14	212	131	53	33	0.8200	0.7736	0.7961
CGRPR	244	8000	61	2000	0.5495	1.0	0.7093

[a]For each set, the number of active and inactive training instances used to build a naïve Bayesian classification model, and the number of active and inactive test compounds are given. In addition, the classification performance is reported. Precision is calculated as the ratio of true active predictions over all active predictions, recall is the ratio of correctly predicted actives over all actives, and the F1-score is the harmonic mean of both.

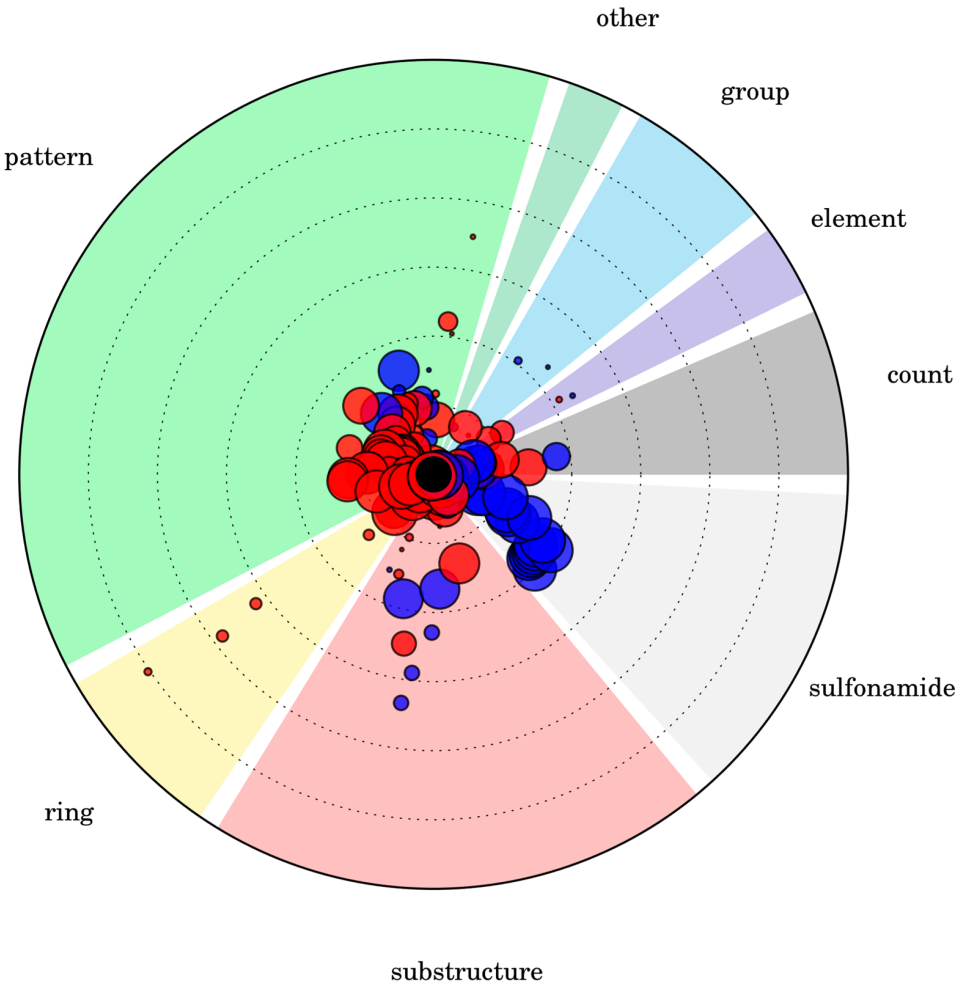


Figure 4. CAI model visualization. All 22 MACCS features associated with the signature sulfonamide are combined into one group. Graphical analysis confirms the hypothesis that the classification model is primarily emphasizing sulfonamide-associated features (see text for further details).

(MAPK14)[34] originating from the ProQinase free choice biochemical assay system.[35] The composition of the data sets is reported in Table 3.

Selected ChEMBL compounds were required to be tested in a direct binding assay with a ChEMBL confidence score of 9. Furthermore, only compounds were considered for which a K_i value of less than 10 μM was available. Using in-house scripts, all candidate compounds were filtered for duplicates, undesired reactive groups, and PAINS[36] liabilities to arrive at a final selection. From compounds in all three data sets, Bemis-Murcko scaffolds (BMS)[37] and corresponding carbon skeletons (CSK),[38] in which all heteroatoms are converted to carbons and all bond orders set to 1, were systematically extracted. Decreasing compound-to-BMS and compound-to-CSK ratios generally indicate increasing structural diversity.

The CAI set was selected because it contained small inhibitors mostly sharing a sulfonamide group (992 of 1306 compounds), which is a hallmark for carbonic anhydrase inhibition. Hence, in this case, a defined chemical moiety was known to be a major determinant of activity. This inhibitor set yielded 407 BMS, 179 CSK, and compound-to-BMS and -CSK ratios of 3.21 and 7.30, respectively.

The MAPK14 set consisted of 429 ATP-site directed kinase inhibitors, 265 of which inhibited the MAP14 kinase. Hence, the remaining compounds were confirmed to be inactive against this kinase (but, in part, active against other kinases). All 429

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

inhibitors contained a conserved pyridinyl-imidazole core with varying substitutions. Hence, this data set was selected as a structurally homogeneous set enabling the derivation of predictive models of MAP14 kinase activity.

The CGRPR set was characterized by a much higher degree of structural diversity than the other two sets, with compound-to-BMS and -CSK ratios of 2.29 and 3.91, respectively, and was therefore selected for our analysis. It should be noted that this compound set had the lowest ratios among 122 structurally heterogeneous candidate sets extracted from ChEMBL for which we have internally built and evaluated naïve Bayesian classification models.

For classification of CAI and CGRPR ligands, a random sample of 10,000 other compounds was taken from ChEMBL to serve as negative training and test instances. For the kinase data set, confirmed inactive compounds were used.

Molecular Representation. MACCS structural keys[39] were used as an exemplary molecular representation. The public version of the MACCS fingerprint consists of a set of 166 structural fragments or patterns, which were generated using an in-house program based upon OpenEye's OEChem toolkit[40] and SMARTS patterns adapted from RDKit.[41] For visualization, MACCS features were organized into different groups:

1. "ring": All ring-related features, e.g., "4M ring" (13 MACCS features),
2. "count": Occurrence count features, e.g., "O > 2" (13 features),
3. "group": Features representing a periodic table group, e.g., "actinide" (11 features),
4. "element": Features representing single specific element, e.g., "P" (9 features),
5. "substructure": Specific substructures, e.g., "ON(C)C" (39 features),
6. "pattern": Substructures with wildcards or exclusions, e.g., "QAAA@1" (76 features),
7. "other": All remaining features (5 features).

Features potentially falling into multiple categories were assigned to a single group in the order of decreasing priority from groups 1−6. For example, the feature "Aromatic Ring >1" was assigned to the "ring" group.

The visualization is also applicable to other types of binary fingerprint representations such as fragment or extended connectivity fingerprints. In the current study, we limit the application to the MACCS fingerprint because of its small size and ease of interpretation. A prototypic implementation of the visualization method is made available (see below), which also provides a basis for further studies with other molecular representations.

Model Building and Evaluation. Models were generated as described in the Concepts and Methods section using the naïve Bayesian formulation with Bernoulli features of the freely available Python machine learning toolkit Scikit-learn.[42] A smoothing factor $\alpha = 1$ and example weights inversely proportional to the class balance were used, which prevented potential smoothing artifacts due to imbalanced data and resulted in assumed uniform prior probabilities. As reported in Table 4, each model was trained on a random subset of 80% of the active compounds. The remaining 20% were used as positive test instances. For CAI and CGRPR, 8000 and 2000 randomly chosen ChEMBL compounds were used as negative training and test examples, respectively. For MAPK14, 80% and 20% of the confirmed inactive compounds were used as negative training and test examples, respectively (Table 4). Hence, the composition of training and test sets for MAPK14 modeling principally differed from CAI and CGRPR.

RESULTS AND DISCUSSION

The principles of *model visualization* and *prediction visualization* are illustrated in Figure 2 and Figure 3, respectively, and have been discussed in the Concepts and Methods section. In the following, we present a number of data set applications to evaluate the visualization techniques in greater detail and analyze models and predictions. First, the prediction performance of the different Bayesian classification models is reported.

Model Performance. In Table 4, the performance of the naïve Bayesian classifiers derived for the three compound data sets is summarized. For CAI and CGRPR from ChEMBL, recall is very high (∼0.91 and 1.00, respectively) but precision only intermediate (∼0.54 and ∼0.55, respectively). For MAPK14, a smaller and more balanced data set, recall performance is lower (∼0.77) but precision higher (0.82) than for the ChEMBL data models, which results in a higher F1-score (∼0.80). Overall, the classifiers derived for compound data sets of different composition and structural complexity display reasonable to high accuracy, a prerequisite for meaningful evaluation of classification models and predictions.

Model Visualization. *CAI.* In Figure 4, the naïve Bayesian model for CAI is visualized. In this case, an additional feature group was defined to which the 22 MACCS features were assigned that are associated with the sulfonamide substructure "*S(=O)(=O)N". Thus, in the scatter plot, all features related to the sulfonamide group are easily identified. Blue coloring of these features confirms that the classification model associates features set in sulfonamide-containing inhibitors with activity. In addition, the size of the corresponding feature points indicates that these features are set in most of the active compounds. While 115 of the 166 MACCS features have an absolute log odds ratio smaller than one, 13 of the 22 sulfonamide features have an

Table 5. Log Odd Ratios and Class Likelihoods of Selected CAI Model Features[a]

feature	group	log odds ratio	class likelihood
4 M ring	ring	−5.0271	0.0169
3 M ring	ring	−3.8505	0.0502
QAAA@1	pattern	−3.4860	0.0036
OS(O)O	substructure	3.3418	0.0844
7 M ring	ring	−3.1828	0.0488
S−O	substructure	2.8918	0.0853
NC(C)N	substructure	−2.4876	0.2433
Si	element	2.3116	0.0049
OQ(O)O	substructure	2.2906	0.0863
group IVa,Va,VIa rows 4−6	group	2.2683	0.0011
C=C(Q)Q	pattern	−2.2216	0.1423
P	element	−2.1154	0.0089
QAA@1	pattern	−2.0519	0.0009
group IIIA (B...)	group	2.0500	0.0164
QQH	sulfonamide	2.0193	0.8227
NS	sulfonamide	2.0088	0.7787

[a]Reported are all MACCS features from the CAI prediction model having an absolute log odds ratio greater than two. If the log odds ratio is negative, the class likelihoods are reported for the negative class; if the log odds ratio is positive, they are reported for the positive class. Log odds ratios and class likelihoods reflect the radius and size of the feature points in Figure 4.

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

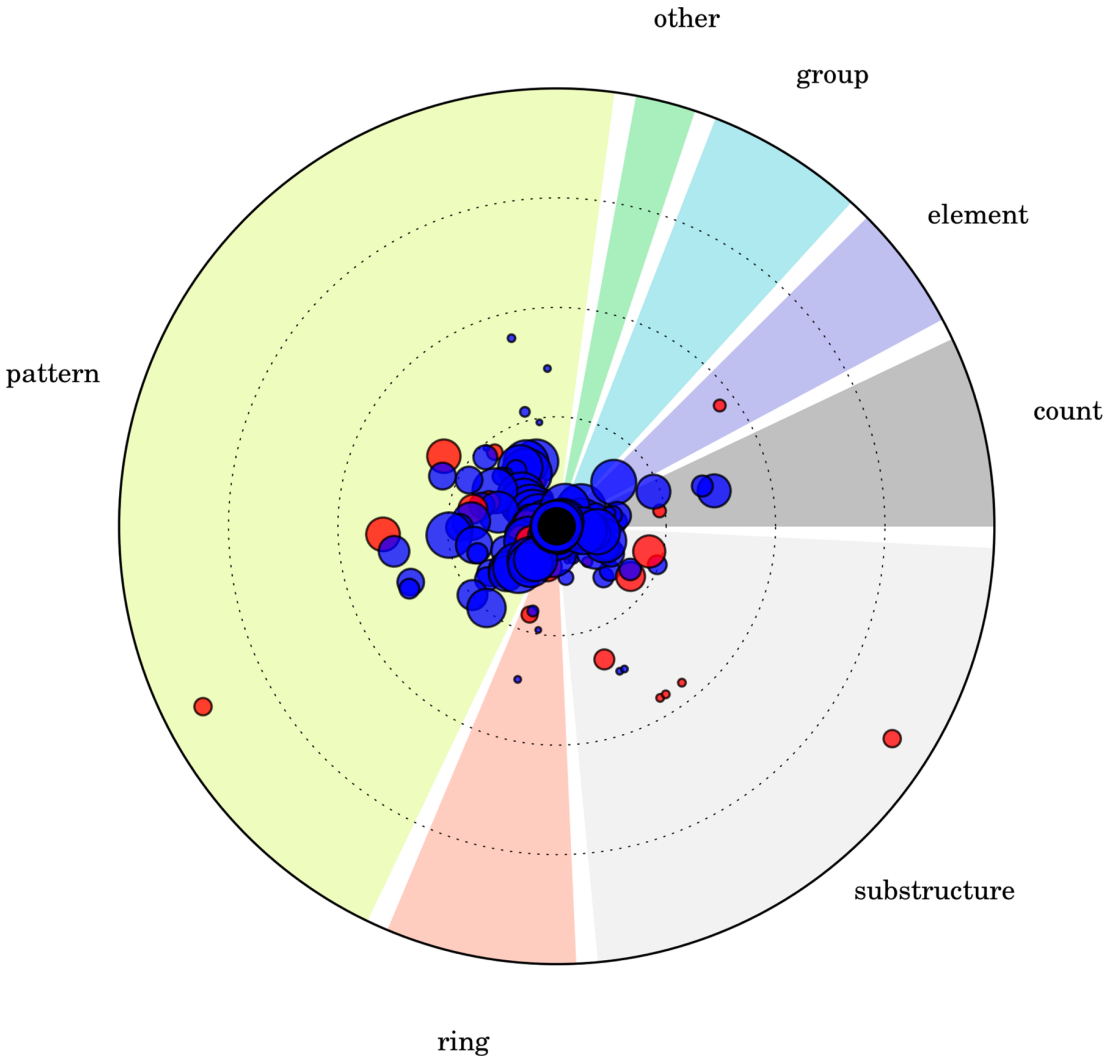


Figure 5. MAPK14 model visualization. Features important for selection of active and deselection of inactive compounds are identified.

absolute log odds ratio larger than one, which confirms their relevance for the model. However, the visualization also reveals that there are other features with much larger absolute log odds ratios than the sulfonamide features, which include three "ring" and three "pattern" features indicating inactivity, two features from the "element" group (one promoting activity and the other inactivity), and four features from the "substructure" group (three indicating activity and one inactivity). All features having an absolute log odds ratio greater than two are summarized in Table 5. However, while the log odds ratio of these features is high, their class likelihood is rather low, as reflected by smaller points. This is indicative of an underrepresentation of these features in the data. Exceptions include the substructure NC(C)N and the pattern C=C(Q)Q, which are present in 24.3% and 14.2% of the inactive compounds, respectively, and have log odds ratios of −2.49 and −2.22. This means that these features are approximately 12 and 10 times more likely to appear in inactive than active compounds. Furthermore, the two sulfonamide features QQH and NS are contained in 82.3% and 77.9% of the active compounds, respectively, and have a log odds ratio of 2.02 and 2.01, indicating that they are approximately 7.5 times more likely to appear in active than inactive compounds.

Taken together, the visualization of the CAI model clearly not only confirms a critically important role of the sulfonamide moiety for the prediction of activity but also demonstrates that there are other features the model regards as even more important for prediction of activity than the sulfonamide. The relatively small class likelihoods of these features point at data imbalance during training, consistent with the limited precision of the model.

MAPK14. The visualization of the MAPK14 model is shown in Figure 5. Here, we again observe that most of the MACCS features have an absolute log odds ratio smaller than one and can hence be considered less important for activity prediction. However, there are a number of features with absolute log odds ratios between one and two, and most of these features promote activity. Finally, a substructure and a pattern feature (C=N and N=A) have absolute log odds ratios larger than three and thus elicit the largest influence on the prediction of activity, provided they are set in the fingerprint of a given compound. Interestingly, both features have a negative log odds ratio, meaning that they are used by the model to deselect inactive compounds, rather than select active ones. By contrast, the features "QCH2A > 1" and "CH3AACH2A" support prediction of activity. All features with an absolute log odds ratio greater than one are listed in Table 6.

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

Table 6. Log Odd Ratios and Class Likelihoods of Selected MAPK14 Model Features[a]

feature	group	log odds ratio	class likelihood
C=N	substructure	−3.6285	0.1091
N=A	pattern	−3.6285	0.1091
BR	element	−1.8536	0.0484
S−O	substructure	−1.8304	0.0181
OS(O)O	substructure	−1.8304	0.0181
OQ(O)O	substructure	−1.8304	0.0181
CH2=A	pattern	1.7671	0.0170
A$A($A)$A	pattern	−1.5894	0.4279
CH3AAACH2A	pattern	1.5042	0.3546
QCH2A > 1 (&...)	count	1.4754	0.4109
CH3AACH2A	pattern	1.4627	0.1436
S heterocycle	ring	1.4436	0.0123
NS	substructure	1.4436	0.0123
CSN	substructure	1.4436	0.0123
C=C(Q)Q	pattern	1.4436	0.0123
CH3ACH2A	pattern	1.4289	0.2655
CH3 > 2 (&...)	count	1.3799	0.1623
N−O	substructure	−1.2919	0.1471
NAAAN	pattern	−1.2141	0.4203
CH2QCH2	pattern	1.1404	0.2702
QHAQH	pattern	1.0849	0.0310

[a]Reported are all MACCS features from the MAPK14 prediction model having an absolute log odds ratio greater than one. Log odds ratios and class likelihoods reflect the radius and size of the feature points in Figure 5.

CGRPR. The CGRPR model is visualized in Figure 6. For this structurally diverse compound set, the visualization of the model notably differs from the others. Here, all of the features that promote activity have a log odds ratio smaller than two, whereas the absolute value of the negative log odds ratios even exceeds six, which corresponds to a more than 400 times higher class likelihood for inactive over active compounds. The features with the highest positive log odds ratios are NC(O)N, the 7-membered ring, and S−S with ORs of 1.85, 1.78, and 1.58, respectively. This corresponds to a class likelihood for active compounds that is 4.8−6.3 times higher than for inactive ones. However, the probability of the count feature "QQ > 1" to be set in the negative class is 429 times higher than its probability to be set in the positive class, and similar values are observed for substructures N−O and NO. Features with an absolute log odds ratio of more than three are summarized in Table 7. It can be seen that features with absolute log odd ratios greater than four occur in only a small fraction of randomly chosen ChEMBL compounds. However, there are also features such as OAOO of A$A!S that appear in more than 10% of all assumed negative instances.

The Bayesian classification model of CGRPR successfully not only recovers all active test compounds but also has only intermediate precision. Visualization of the model reveals that nearly all features with a large absolute log odds ratio promote the prediction of inactivity, provided they are present in a compound. This indicates that the model primarily deprioritizes inactive compounds instead of prioritizing active ones. This observation is consistent with the fact that many active compounds in this data set are structurally diverse and cannot be easily distinguished from negative instances by only a few descriptive features. Instead, the model focuses on features that predominantly occur in inactive compounds.

Table 7. Log Odd Ratios and Class Likelihoods of Selected CGRPR Model Features[a]

feature	group	log odds ratio	class likelihood
QQ > 1 (&...)	count	−6.0607	0.0520
N−O	substructure	−6.0139	0.0496
NO	substructure	−6.0139	0.0496
4 M ring	ring	−4.9351	0.0169
QHQH (&...)	pattern	−4.9277	0.0167
QCH2Q	pattern	−4.8661	0.0157
P	element	−4.2923	0.0089
OQ(O)O	substructure	−4.2781	0.0087
I	element	−4.2637	0.0086
CH2=A	pattern	−3.9807	0.0065
OAAO	pattern	−3.7200	0.1741
ON(C)C	substructure	−3.7182	0.0050
A$A!S	pattern	−3.5099	0.1411
CQ(C)(C)A	pattern	−3.4631	0.0039
QAAA@1	pattern	−3.3963	0.0036
NS	substructure	−3.2094	0.1045
OS(O)O	substructure	−3.2069	0.0030
CSN	substructure	−3.1716	0.1006
QHAAAQH	pattern	−3.1219	0.0957

[a]Reported are all MACCS features from the CGRPR prediction model having an absolute log odds ratio greater than three. All log odds ratios are negative indicating that all features support prediction of inactivity.

Model Characteristics. Taken together, model visualizations for the three compound data sets reveal the presence of different model characteristics. The CAI model primarily prioritizes compounds containing the sulfonamide signature (as to be expected) and deprioritizes compounds with specific ring systems or patterns. The MAPK14 selects compounds on the basis of specific features that are preferentially set in active compounds. By contrast, the CGRPR model primarily deselects inactive compounds instead of prioritizing actives. Hence, classification models derived for data sets of varying composition and structural complexity display different model characteristics. Graphical analysis clearly reveals key feature for predictions using the different models.

Feature Selection. Another interesting application of model visualization is the rationalization of feature selection effects. To illustrate this point, we have subsequently removed the features with the highest log ORs from our models and monitored the change in model performance (data not shown). Removal of the first few features from the CAI model did not alter performance significantly. This might seem surprising at first glance. However, Figure 4 shows that features with highest log OR were only very infrequently set. Therefore, removal of these features did not influence the majority of new predictions. By contrast, when features with a higher probability to be set were removed, for instance, the larger circles from the "substructure" and "sulfonamide" areas, classification performance changed significantly. Equivalent observations were made for the MAPK14 model. Removal of features with high log odd ratios only slightly affected predictive performance, if these features were only rarely set. Removal of additional features resulted in further improved recall but reduced precision−a direct consequence of predicting more compounds as active. This effect can also be rationalized by analyzing Figure 5 where the outermost features were responsible for compound deselection. Finally, removing the most important features from the CGRPR model resulted in reduced precision, which can also be attributed to the fact that

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

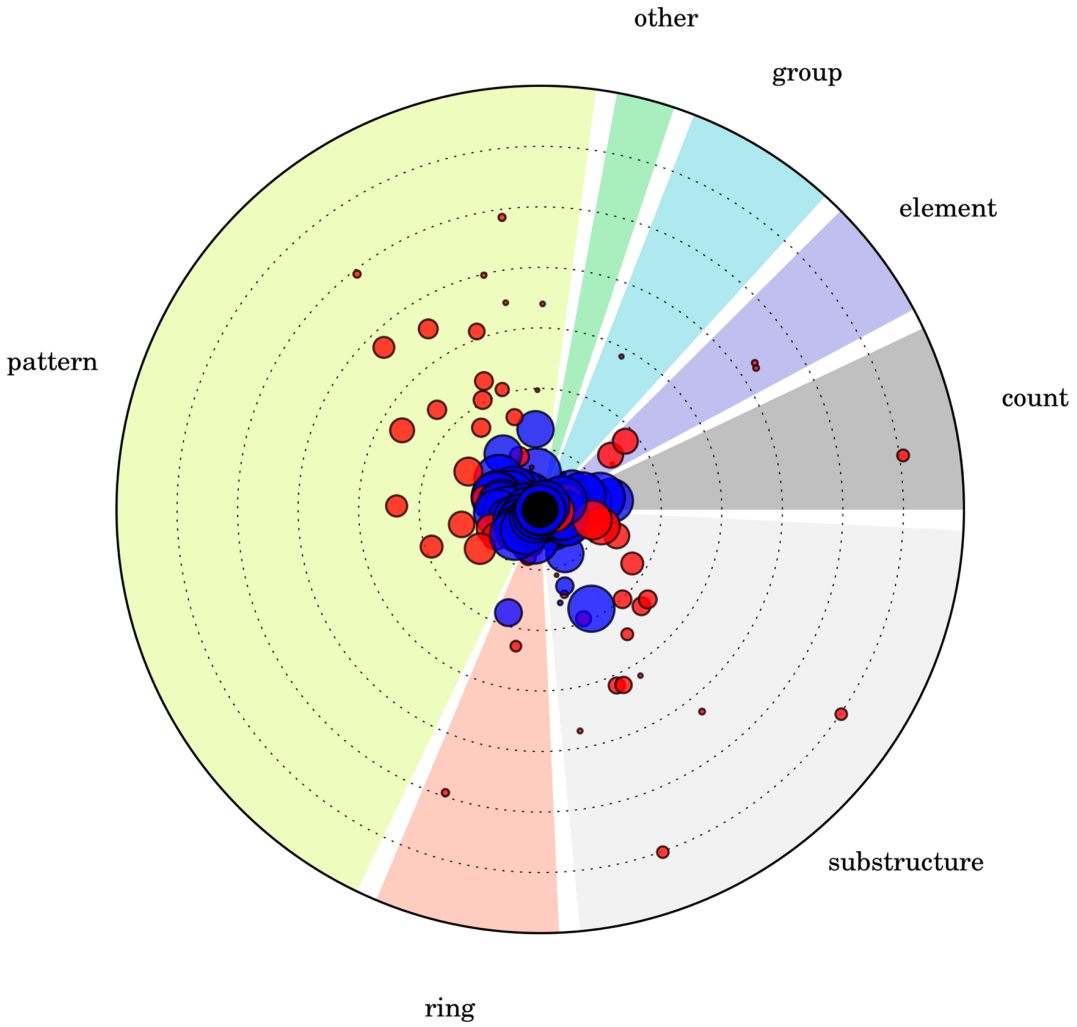


Figure 6. CGRPR model visualization. The prediction model for the structurally diverse CGRPR class strongly deselects inactive compounds based on features with negative log odd ratios (red circles).

only deselecting features were removed. Clearly, overall prediction performance is predominantly affected by distinguishing features that are frequently set in compounds, which can be well appreciated on the basis of model visualization.

Prediction Visualization. It is important to note that the model visualization emphasizes log odds ratios given features are set in compound fingerprints, i.e., $P(x_d = 1|y = 1)/(P(x_d = 1|y = 0))$ (cf. eq 8). However, if a feature is not set in the fingerprint of a compound, the term entering the classification rule is changed to

$$\frac{P(x_d = 0|y = 1)}{P(x_d = 0|y = 0)} = \frac{1 - P(x_d = 1|y = 1)}{1 - P(x_d = 1|y = 0)} \quad (14)$$

This means that features with a high log odds ratio given their presence can have low log odds ratios in their absence, which can further complicate the understanding of model decisions. Hence, to better understand individual predictions, rather than global model performance, a *prediction visualization* method has also been introduced.

Table 8 summarizes true and false positive and negative predictions for the test sets of the three models. We will use compounds from these different subsets as examples for prediction visualization.

Table 8. True or False Positive and Negative Predictions[a]

data set		no. of cpds predicted to be active	no. of cpds predicted to be inactive
CAI	no. of active compounds	238	24
	no. of inactive compounds	206	1794
MAPK14	no. of active compounds	41	12
	no. of inactive compounds	9	24
CGRPR	no. of active compounds	61	0
	no. of inactive compounds	50	1950

[a]For each model, the number of true positives (correctly predicted active compounds), true negatives (correctly predicted inactive compounds), false positives (inactive compounds predicted to be active), and false negatives (active compounds predicted to be inactive) is reported.

CAI. Exemplary true and false positive predictions of the CAI model are visualized in Figure 7. In the fingerprint of the correctly predicted active compound, 53 of 166 features are set, 33 of which have a positive and 20 a negative log odds ratio. However, 42 and 71 of the 113 MACCS substructures not present in the fingerprint have a positive and negative log odds ratio, respectively. Considering eq 14, this means that the 71 features that are not set and have a negative log odds ratio in the

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

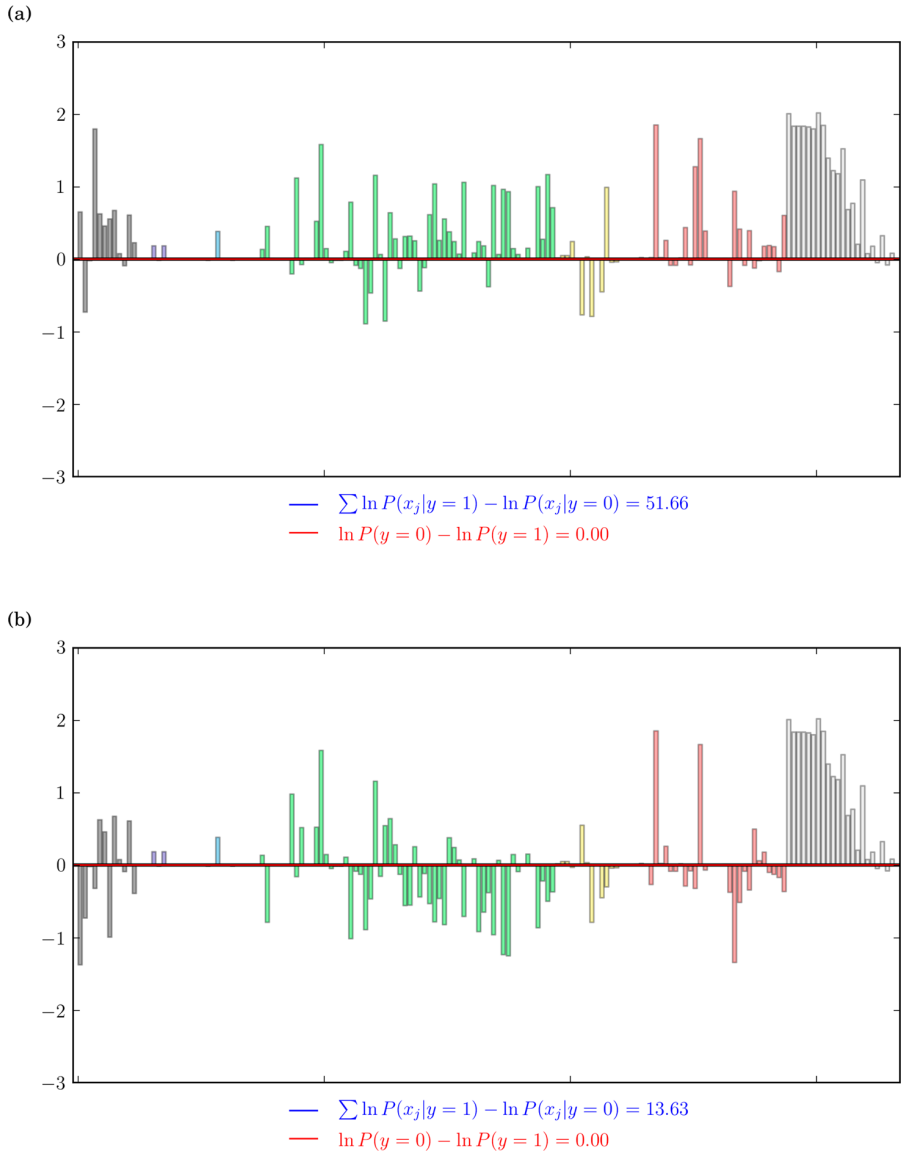


Figure 7. Prediction visualization for CAI. Shown is the prediction visualization for two compounds predicted to be active and representing a (a) true positive and (b) false positive, respectively.

global model actually make a positive contribution to the prediction of activity for this compound. In total, the presence or absence of 104 (33 plus 71) of 166 MACCS features contributes to the prediction of activity, whereas the remaining 62 support prediction of inactivity. The actual weight of their respective contributions is depicted in Figure 7a. It is evident that most of the negative contributions are small (none of them exceeds −1). On the other hand, many of the positive contributions fall in the range between +1 to +2. In total, the sum of positive log class likelihoods is 51.66, which results in a clear positive prediction, considering that the smallest ratio of class log priors to be exceeded for a positive prediction is 0. Taking the group coloring in Figure 4 into account, one can immediately conclude that the most positive contributions result from the sulfonamide group (light gray), followed by the pattern (green), the substructure (red), and the count group (dark gray). By contrast, significant negative contributions come from the feature "Heterocyclic atom >1" (count group; dark gray), the ring features "5 M ring" and "N Heterocycle" (yellow), and the patterns "NAAN" and "QAAAA@1" (green). According to the model, these features are more likely to be set by inactive compounds, but the presence or absence of features that support prediction of activity outweighs these contributions.

Figure 7b shows the visualization for a false positive prediction by the CAI model. In this case, there are more negative contributions from different feature groups than for the example in Figure 7a. Yet, the sum of log odd ratios still is 13.63, giving rise to a positive prediction. The features with the largest influence on this prediction include the patterns "A$A!S" and "SA(A)A" (green), the substructures "CSN" and "CSO" (red), and most of the features associated with the sulfonamide group (light gray). The latter contributions are largely responsible for the false positive prediction, although other infrequently observed structural features render this compound inactive. This reflects a limitation of the model for predicting compounds that contain the sulfonamide but are nonetheless inactive for other reasons

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

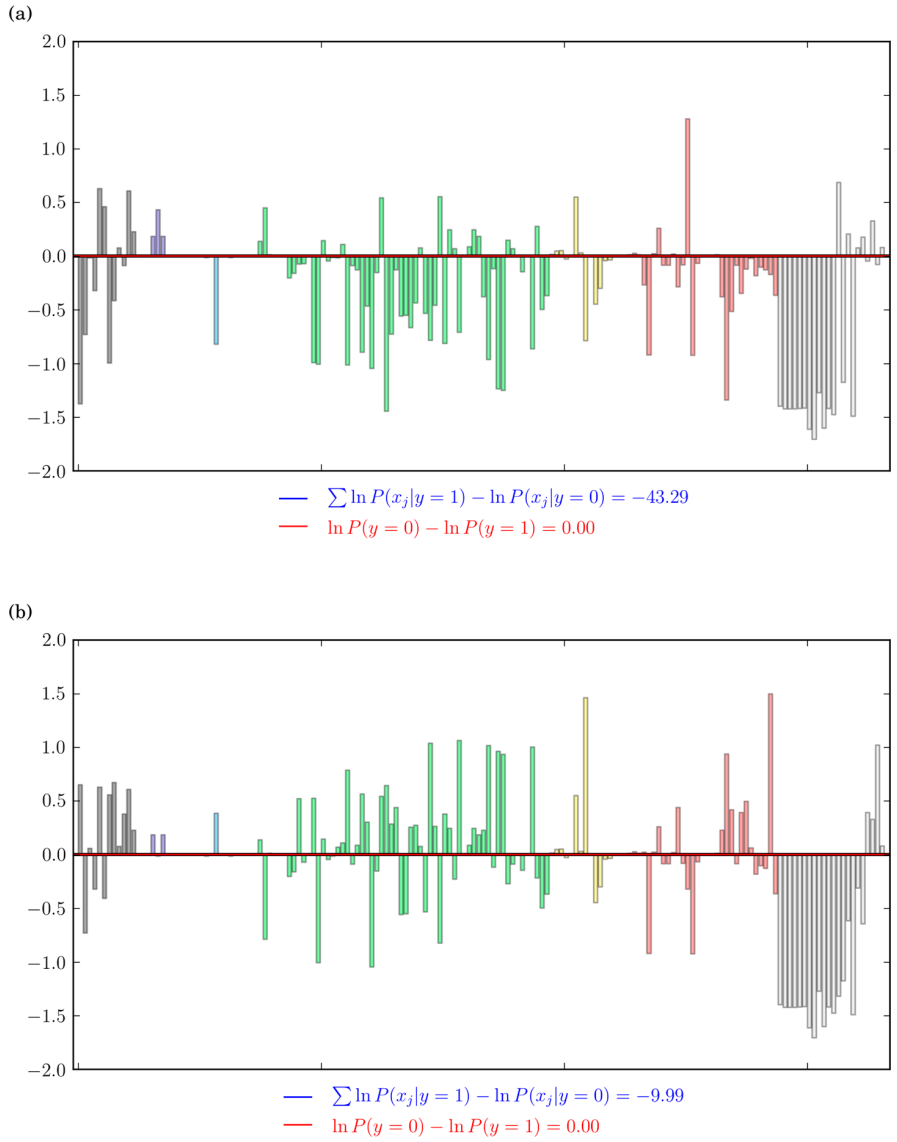


Figure 8. Prediction of inactivity for CAI. Shown is the prediction visualization for two compounds predicted to be inactive and representing a (a) true negative and (b) false negative, respectively.

(such as, for example, steric hindrance due to the presence of other groups).

Exemplary true and false negative CAI predictions are visualized in Figure 8. For the correctly predicted inactive compound, many negative contributions are observed resulting in a sum of class log likelihoods of −43.29 (Figure 8a). By contrast, the active compound has a more balanced ratio of both positive and negative contributions resulting in a sum of class log likelihoods of −9.99 (Figure 8b). Although many positive contributions are detected for the active compound, the negative prediction is mostly due to the absence of a sulfonamide group in this compound, which again reflects the focus of the classifier on this signature group shared by the majority of active compounds. Features associated with this group (light gray) make the strongest negative contribution due to their absence.

MAPK14. Figure 9a visualizes a true positive prediction by the MAPK14 model. Here, both positive and negative contributions of the count group (dark gray), the pattern group (yellow), and the substructure group (light gray) become apparent. Features resulting from single elements (purple), groups (blue), or rings (red) only make minor positive contributions to the class likelihood. In total, however, the positive contributions outweigh the negative ones; hence, the compound is predicted to be active. The largest positive contributions come from the presence of the patterns "QHAQH", "S=A", and "QA(Q)Q" and the largest negative terms from the absence of the patterns "AQ(A)A" and "QCH2A". Overall, this prediction clearly reflects the presence of a cumulative effect of many small-magnitude contributions accounted for by the model. Figure 9b visualizes a true negative prediction using this model, which helps to better understand its strong tendency to deselect inactive compounds, as discussed above. There are positive contributions from count features, substructures, and patterns, but the magnitude of negative contributions is by far larger. The absence of a sulfur atom (purple) and of the patterns "A!N$A", "QA(Q)Q", and "AN(A)A" (yellow) have the strongest negative influence. In fact, most influential terms come from the absence of substructural features. Hence, the finding from the global

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

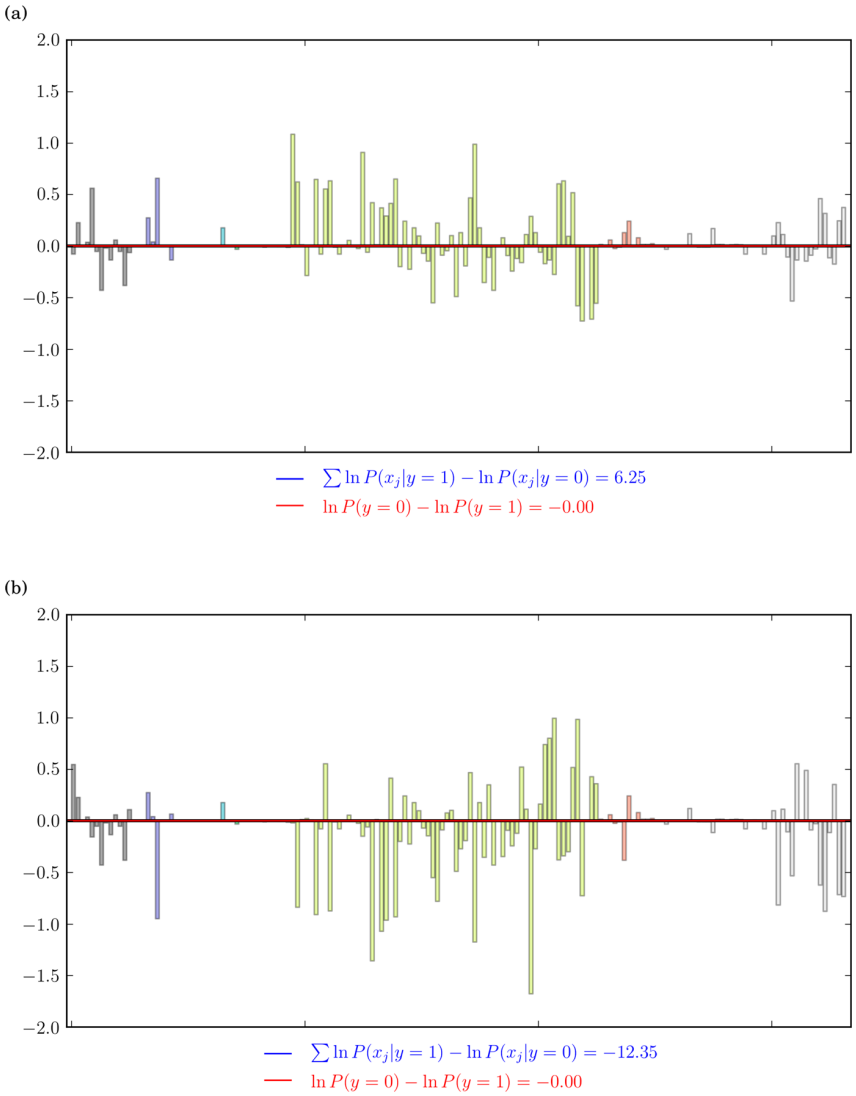


Figure 9. Correct predictions for MAPK14. Shown is the prediction visualization for two compounds representing a (a) true positive and (b) true negative, respectively.

model analysis that the MAPK14 model preferentially deselects inactive compounds is further substantiated at the level of individual predictions.

CGRPR. Figure 10 analyzes active and inactive compounds correctly predicted by the CGRPR model, respectively. For the true positive prediction in Figure 10a, there are a four major negative terms in the log likelihood sum accounting for the absence of the pattern "C$=C($A)$A" and the substructure "C═CN" and for the presence of the patterns "QHAAQH" and "QHAQH". On the other hand, there are two peaks indicating positive contributions including the 7-membered ring (red) and the substructure "NC(O)N" (light gray), which are both set in the fingerprint of this compound. The remaining positive contributions are comparably small. Hence, in this case, there also is a cumulative effect leading to the prediction of activity for this compound. This is consistent with the structural heterogeneity of the CGRPR set and the absence of simple structural rules that distinguish active from inactive compounds (such as the presence or absence of specific functional groups).

In Figure 10b, a completely different picture emerges. Here, the majority of features in the fingerprint have a positive log odds ratio, although all of these contributions are of rather low magnitude. In addition, there are a few small and medium-size contributions to negative class log likelihood, with three major feature peaks. These include the absence of patterns "NAN" and "QHAAACH2A" and of the substructure "NH". In other words, the model does not strictly require any active compound to possess these features but strongly deprioritizes compounds that do not have these features set in their fingerprints.

The visualization of the inactive prediction in Figure 10b also illustrates another important aspect: While the model visualization in Figures 4−6 can highlight features that (according to the model) make significant contributions to the prediction of activity or inactivity, it cannot fully represent the decision process for any new test instance. For example, Figure 6 indicates that features "QQ > 1″, "N−O", and "NO" might dominate the predictions. However, this is only the case if these features are present in a test compound. The prediction for compounds where these features are not set, as the one depicted in

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

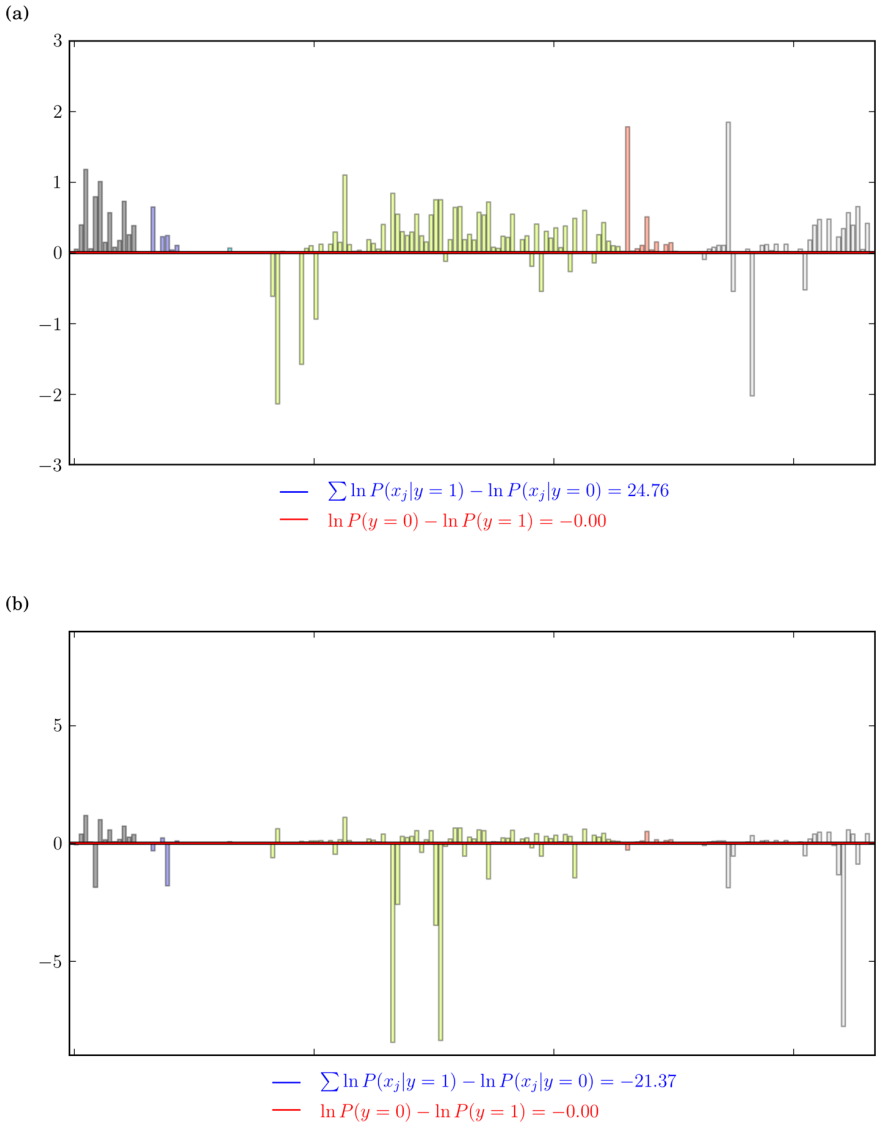


Figure 10. Correct predictions for CGRPR. Shown is the prediction visualization for two compounds representing a (a) true positive and (b) true negative, respectively.

Figure 10b, is hardly influenced by any of them. For example, let us consider the feature "QQ > 1". If it is present, its log odds ratio is given by

$$\log \frac{P(x_{QQ>1} = 1|y = 1)}{P(x_{QQ>1} = 1|y = 0)} = \log \frac{0.0001}{0.0520} = -6.0607$$

However, if it is absent, the log odds ratio approaches zero:

$$\log \frac{P(x_{QQ>1} = 0|y = 1)}{P(x_{QQ>1} = 0|y = 0)} = \log \frac{1 - P(x_{QQ>1} = 1|y = 1)}{1 - P(x_{QQ>1} = 1|y = 0)} = \log \frac{0.9999}{0.9480} = 0.0533$$

In this case, other features play a by far more important role. It follows that both a careful analysis of global model performance as well as of individual predictions is required to fully rationalize why classification models might yield—or not yield—accurate predictions.

Feature Mapping. Depending on the chosen molecular descriptors, features that are most important for a prediction can be back-projected onto test compounds, which aids in the exploration of SARs. Fragment fingerprints such as MACCS are suitable descriptors for feature mapping. Figure 11 shows examples of correctly predicted active and inactive compounds and of key features that are present in these compounds and make major contributions to the prediction of activity or inactivity. For each compound, only features with a log odds ratio of at least 90% of the maximum or minimum OR are mapped. These features are reported in Table 9. For example, features of the compound in Figure 11a have a minimum and maximum log odds ratio of −0.89 and 2.02, respectively. Requiring at least 90% of this value, we highlight features with log odds ratios smaller than 0.8 or greater than 1.82. By contrast, no color-coding can be applied for the compound in Figure 11b because it is predicted to be inactive since it is missing essential substructures. This illustrates limitations of feature mapping approaches. The examples in Figure 11 reveal that features that are present in

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

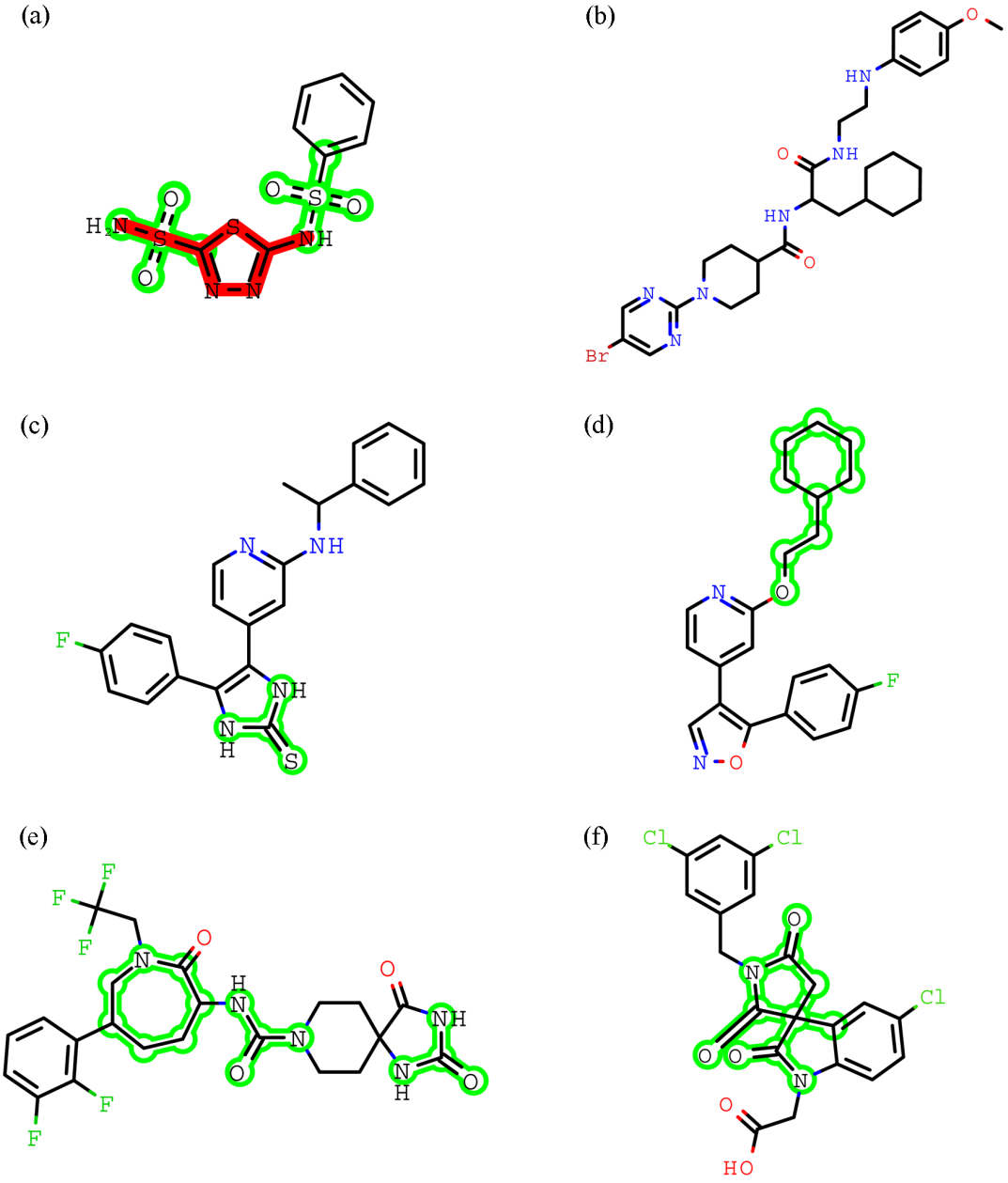


Figure 11. Feature mapping. Features with at least 90% log odds ratio compared to the maximum or minimum are back-projected onto test compounds. Shown are correctly predicted active and inactive compounds for which predictions are visualized in Figures 7−10: (a) active/CAI, (b) inactive/CAI, (c) active/MAPK14, (d) inactive/MAPK14, (e) active/CGRPR, and (f) inactive/CGRPR. Red and green coloring indicates features with negative and positive log odd ratios, respectively. The depiction of the feature mapping was created using OpenEye's OEDepict Toolkit.[43]

compounds and determine predictions cover substructures of different size. However, feature mapping only provides an incomplete account of predictions, in contrast to prediction visualization, because of the frequent importance of feature absence, as revealed in our analysis.

Figure 12 shows two active and two inactive compounds from the CAI test set. The two compounds at the top were correctly predicted as active, whereas the bottom left compound represented a true negative and the bottom right compound a false positive prediction. Color shading indicates the magnitude of the mapped features' log odds ratio (only features with a log OR of at least 25% of the maximum absolute log OR were considered for mapping). Interestingly, the compounds predicted to be active also contain a rather large red area, i.e., features that the model utilized to deselect inactives. However, nonset features that cannot be mapped often had a major influence on the predictions. For instance, the sum of log ORs of the features that are set in the compound in Figure 12a is −0.14. This means that this compound would be predicted to be inactive if the prediction would only be based on mapped substructures. However, the sum of log ORs of the nonset features is 2.52, which hence leads to the prediction of activity. The other two compounds predicted to be active have a log OR sum of 5.63 and 23.94 for their set features, and 6.77 and 22.08 for their nonset features, respectively. The true negative prediction in Figure 12c has an overall log OR sum of −3.8 (with both the sum of the set

dx.doi.org/10.1021/ci500410g | J. Chem. Inf. Model. 2014, 54, 2451−2468

Table 9. Features Most Important for Individual Predictions[a]

compound	feature	presence	OR
(a)	CSN	+	1.85
	NS	+	2.01
	OSO	+	1.84
	QSQ	+	1.84
	S=O	+	1.84
	AS(A)A	+	1.82
	QQH	+	2.02
	S=A	+	1.85
	NAAN	+	−0.89
	QAAAA@1	+	−0.85
(b)	C=C(C)C	-	1.28
	QQH	-	−1.61
	S=A	-	−1.71
	S	-	−1.60
(c)	QHAQH	+	1.08
	QA(Q)Q	+	0.99
	AQ(A)A	-	−0.73
	QCH2A	-	−0.71
(d)	AN(A)A	-	−1.68
	OACH2A	+	0.99
	ACH2CH2A	+	0.98
(e)	7 M ring	+	1.78
	C$=C($A)$A	-	−2.14
	NC(O)N	+	1.85
	C=CN	-	−2.03
(f)	CC(C)(C)A	+	1.10
	NAN	-	−8.45
	QHAAACH2A	-	−8.38
	A$A!O > 1	+	1.18
	NH	-	−7.78

[a]Listed are the most important features for the prediction of the compounds shown in Figure 11. Features that are present can be mapped, whereas the influence of features that are absent can only be inferred from the prediction visualization. The odds ratios are reported for feature presence or absence, i.e., $OR_d = P(x_d = 1 \mid y = 1)/P(x_d = 1 \mid y = 0)$ for present and $OR_d = P(x_d = 0 \mid y = 1)/P(x_d = 0 \mid y = 0)$ for absent features.

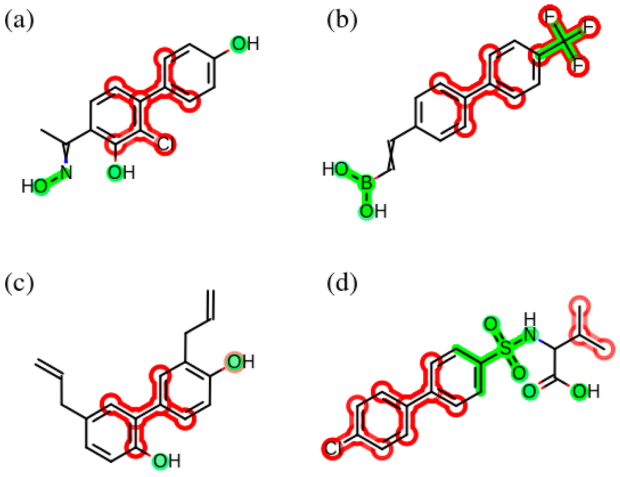


Figure 12. Feature mapping of selected CAI compounds. Features with at least 25% log odds ratio compared to the maximum absolute log OR are back-projected onto four selected CAI compounds including a (a) true positive, (b) true positive, (c) true negative, and (d) false positive. Color shading indicates the magnitude of the log odd ratios.

(−2.13) and the nonset (−1.68) features being negative). Hence, in this case, the influence of set and nonset features is of comparable magnitude.

CONCLUDING REMARKS

In this work, a visualization approach for Bayesian classification models and their predictions has been introduced. Naïve Bayesian classifiers are widely used in chemoinformatics. Although Bayesian classification is methodologically less complex than other machine learning approaches such as support vector machines or neural networks, analyzing classification models and rationalizing their performance are far from being trivial tasks. Features that determine the performance of Bayesian models and their potential interplay are difficult to identify, especially if classification proceeds in high-dimensional reference spaces, which is usually the case. In a few instances in which it has thus far been attempted to rationalize the performance of machine learning models, statistical considerations have been applied, for example, in the context of feature selection. Others have used visualization schemes in terms of feature mapping, where they could by design only account for present but not absent features. We have designed a new graphical analysis scheme for "model anatomy" and demonstrated the utility of model visualization and prediction visualization to better understand how Bayesian classification models work. Exemplary compound data sets of different composition and structural heterogeneity, with known or unknown SAR determinants, were used to build Bayesian classification models for activity prediction. On the basis of graphical analysis, we have been able to determine that classification models respond differently to structural characteristics of these compound sets and that feature absence and deselection of inactive compounds often contributes as much (or even more) to prediction accuracy as feature presence and preferential selection of active compounds. The identification of signature features and/or cumulative feature effects play comparably important roles for global model performance and individual predictions. Graphical analysis of the CAI model and representative predictions has demonstrated how the visualization approach introduced herein helps to rationalize model performance and focus on key features. For the more complex data sets MAPK14 (containing kinase inhibitors that are structurally very similar to inactives) and CGRPR (with high structural heterogeneity among actives), the visualizations have enabled us to better understand why classification models reach reasonable to good predictive performance even in these rather difficult cases. Here, our findings highlight the role of compound deselection and cumulative feature effects referred to above. Taken together, our results suggest that model visualization, as introduced herein, should aid in the rationalization and further refinement of Bayesian classification methods. The visualization approach should also be adaptable for other supervised machine learning methods and help reduce their often cited "black box" character. A prototypic Python implementation of our visualization methodology is made freely available via the public Zenodo platform.[44] This implementation should provide a basis for further exploration and extension of our visualization approach.

AUTHOR INFORMATION

Corresponding Author

*Phone: +49-228-2699-306. Fax: +49-228-2699-341. E-mail: bajorath@bit.uni-bonn.de.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Ye Hu and Norbert Furtmann for help with data collection and feature analysis. The use of OpenEye's OEChem and OEDepict Toolkit was made possible by their free academic licensing program.

■ REFERENCES

(1) Varnek, A.; Baskin, I. Machine Learning Methods for Property Prediction in Chemoinformatics: *Quo Vadis*? *J. Chem. Inf. Model.* **2012**, *52*, 1413−1437.
(2) Vogt, M.; Bajorath, J. Chemoinformatics: A View of the Field and Current Trends in Method Development. *Bioorg. Med. Chem.* **2012**, *20*, 5317−5323.
(3) Geppert, H.; Vogt, M.; Bajorath, J. Current Trends in Ligand-based Virtual Screening: Molecular Representations, Data Mining Methods, New Application Areas, and Performance Evaluation. *J. Chem. Inf. Model.* **2010**, *50*, 205−216.
(4) Ballester, P. J.; Mitchell, J. B. O. A Machine Learning Approach to Predicting Protein-Ligand Binding Affinity with Applications to Molecular Docking. *Bioinformatics* **2010**, *26*, 1169−1175.
(5) Frank, E.; Bouckaert, R. R. Naive Bayes for Text Classification with Unbalanced Classes. *Proc. 10th European Conf. on Principle and Practice of Knowledge Discovery in Databases* **2006**, 503−510.
(6) Heikamp, K.; Hu, X.; Yan, A.; Bajorath, J. Prediction of Activity Cliffs Using Support Vector Machines. *J. Chem. Inf. Model.* **2012**, *52*, 2354−2365.
(7) Hert, J.; Willett, P.; Wilton, D. J. New Methods for Ligand-Based Virtual Screening: Use of Data Fusion and Machine Learning To Enhance the Effectiveness of Similarity Searching. *J. Chem. Inf. Model.* **2006**, *46*, 462−470.
(8) Prince, S. J. D. *Computer Vision: Models, Learning, and Inference*; Cambridge University Press: 2012.
(9) Wawer, M.; Lounkine, E.; Wassermann, A. M.; Bajorath, J. Data Structures and Computational Tools for the Extraction of SAR Information from Large Compound Sets. *Drug Discovery Today* **2010**, *15*, 630−639.
(10) Wassermann, A. M.; Wawer, M.; Bajorath, J. Activity Landscape Representations for Structure-Activity Relationship Analysis. *J. Med. Chem.* **2010**, *53*, 8209−8223.
(11) Stumpfe, D.; Bajorath, J. Methods for SAR Visualization. *RSC Adv.* **2012**, *2*, 369−378.
(12) Whitesides, G. M.; Krishnamurthy, V. M. Designing Ligands to Bind Proteins. *Q. Rev. Biophys.* **2005**, *38*, 385−395.
(13) Stumpfe, D.; Bajorath, J. Similarity Searching. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1*, 260−282.
(14) Glick, M.; Klon, A. E.; Acklin, P.; Davies, J. W. Enrichment of Extremely Noisy High-Throughput Screening Data Using a Naïve Bayes Classifier. *J. Biomol. Screening* **2004**, *9*, 32−36.
(15) Glick, M.; Jenkins, J. L.; Nettles, J. H.; Hitchings, H.; Davies, J. W. Enrichment of High-Throughput Screening Data with Increasing Levels of Noise Using Support Vector Machines, Recursive Partitioning, and Laplacian-Modified Naive Bayesian Classifiers. *J. Chem. Inf. Model.* **2006**, *46*, 193−200.
(16) Glick, M.; Davies, J. W.; Jenkins, J. L. Prediction of Biological Targets for Compounds Using Multiple-Category Bayesian Models Trained on Chemogenomics Databases. *J. Chem. Inf. Model.* **2006**, *46*, 1124−1133.
(17) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of Kinase Inhibitors Using a Bayesian Model. *J. Med. Chem.* **2004**, *47*, 4463−4470.
(18) Rogers, D.; Brown, R. D.; Hahn, M. Using Extended-Connectivity Fingerprints with Laplacian-Modified Bayesian Analysis in High-Throughput Screening Follow-Up. *J. Biomol. Screening* **2005**, *10*, 682−686.
(19) Sutherland, J. J.; Higgs, R. E.; Watson, I.; Vieth, M. Chemical Fragments as Foundations for Understanding Target Space and Activity Prediction. *J. Med. Chem.* **2008**, *51*, 2689−2700.
(20) Wassermann, A. M.; Kutchukian, P. S.; Lounkine, E.; Luethi, T.; Hamon, J.; Bocker, M. T.; Malik, H. A.; Cowan-Jacob, S. W.; Glick, M. Efficient Search of Chemical Space: Navigating from Fragments to Structurally Diverse Chemotypes. *J. Med. Chem.* **2013**, *56*, 8879−8891.
(21) Klon, A. E.; Glick, M.; Davies, J. W. Application of Machine Learning To Improve the Results of High-Throughput Docking Against the HIV-1 Protease. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2216−2224.
(22) Klon, A. E.; Glick, M.; Thoma, M.; Acklin, P.; Davies, J. W. Finding More Needles in the Haystack: A Simple and Efficient Method for Improving High-Throughput Docking Results. *J. Med. Chem.* **2004**, *47*, 2743−2749.
(23) Klon, A. E.; Glick, M.; Davies, J. W. Combination of a Naive Bayes Classifier with Consensus Scoring Improves Enrichment of High-Throughput Docking Results. *J. Med. Chem.* **2004**, *47*, 4356−4359.
(24) Klon, A. E.; Lowrie, J. F.; Diller, D. J. Improved Naïve Bayesian Modeling of Numerical Data for Absorption, Distribution, Metabolism and Excretion (ADME) Property Prediction. *J. Chem. Inf. Model.* **2006**, *46*, 1945−1956.
(25) Sun, H. A Naïve Bayes Classifier for Prediction of Multidrug Resistance Reversal Activity on the Basis of Atom Typing. *J. Med. Chem.* **2005**, *48*, 4031−4039.
(26) Bender, A.; Mussa, H. Y.; Gill, G. S.; Glen, R. C. Molecular Surface Point Environments for Virtual Screening and the Elucidation of Binding Patterns (MOLPRINT 3D). *J. Med. Chem.* **2004**, *47*, 6569−6583.
(27) Nigsch, F.; Bender, A.; Jenkins, J. L.; Mitchell, J. B. O. Ligand-Target Prediction Using Winnow and Naive Bayesian Algorithms and the Implications of Overall Performance Statistics. *J. Chem. Inf. Model.* **2008**, *48*, 2313−2325.
(28) Rosenbaum, L.; Hinselmann, G.; Jahn, A.; Zell, A. Interpreting Linear Support Vector Machine Models with Heat Map Molecule Coloring. *J. Cheminf.* **2011**, *3*, 11.
(29) Riniker, S.; Landrum, G. A. Similarity Maps - a Visualization Strategy for Molecular Fingerprints and Machine-Learning Methods. *J. Cheminf.* **2013**, *5*, 43.
(30) Alpaydin, E. *Introduction to Machine Learning*, 2nd ed.; MIT Press: Cambridge, USA, 2010.
(31) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*, 2nd ed.; Wiley-Interscience: New York, 2000.
(32) Zhang, H. The Optimality of Naïve Bayes. *Proc. 17th Int. Florida Artific. Intell. Res. Soc. Conf.* **2004**, 562−567.
(33) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−D1107.
(34) Dimova, D.; Iyer, P.; Vogt, M.; Totzke, F.; Kubbutat, M. H. G.; Schächtele, C.; Laufer, S.; Bajorath, J. Assessing the Target Differentiation Potential of Imidazole-Based Protein Kinase Inhibitors. *J. Med. Chem.* **2012**, *55*, 11067−11071.
(35) ProQinase Free Choice Biochemical Kinase Assays. http://www.proqinase.com/ (accessed Oct 15, 2013).
(36) Baell, J. B.; Holloway, G. A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *J. Med. Chem.* **2010**, *53*, 2719−2740.
(37) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.
(38) Xu, Y.-J.; Johnson, M. Algorithm for Naming Molecular Equivalence Classes Represented by Labeled Pseudographs. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 181−185.
(39) *MACCS Structural keys*; Accelrys: San Diego, CA, 2011.
(40) *OEChem TK version 2.0.0*; OpenEye Scientific Software: Santa Fe, NM. http://www.eyesopen.com (accessed July 5, 2014).
(41) *RDKit: Open-source cheminformatics*. http://www.rdkit.org (accessed July 5, 2014).

(42) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(43) *OEDepict TK version 2.0.0*; OpenEye Scientific Software: Santa Fe, NM. http://www.eyesopen.com (accessed July 5, 2014).
(44) Balfer, J.; Bajorath, J. Visualization and Graphical Interpretation of Bayesian Compound Classification Models. http://dx.doi.org/10.5281/zenodo.11371.

Summary

In this chapter, we have introduced a method for the visualization of naïve Bayes classification models and individual predictions. The method is suitable for binary fingerprint representations, and a prototypic implementation is made freely available under the DOI `10.5281/zenodo.11371`.

The model visualization takes into account the probability of a feature to occur in a class, the sign and the magnitude of a feature's log odds ratio. It then provides an intuitive explanation of the features that are prioritized by the model. Individual predictions can be visualized to better understand why a certain compound was predicted active or inactive, respectively. The approach was illustrated using three different data sets, and it was shown that the naïve Bayes classifier works not only by prioritizing, but also by deprioritizing compounds based on certain features. Furthermore, a backprojection onto the molecular graph is possible.

The visualization of naïve Bayes classification models is the first contribution towards intuitively interpretable machine learning models for drug discovery. In the next chapter, a similar approach for the visualization of SVM predictions using the Tanimoto kernel is introduced.

6

Visualization and Interpretation of Support Vector Machine Activity Predictions

Introduction

In the last chapter, an interactive visualization for naïve Bayes classification models and predictions was introduced. In addition to naïve Bayes, SVM modeling has extensively been used throughout this thesis. The success of SVMs in drug discovery applications motivated the development of a visualization method for SVM activity predictions.

However, in contrast to naïve Bayes, SVMs are "black box" models and hard to interpret. This is due to their formalization in dual space and the use of kernels. Therefore, it is not possible to derive a general model visualization in terms of input features, as was done for naïve Bayes models in the previous chapter. Instead, we provide a visualization of individual SVM predictions using the linear or Tanimoto kernel on fingerprints. The approach is investigated on different data sets, and the differences between both kernels are highlighted. Furthermore, a mapping of the features onto the molecular graph is used to make the results chemically accessible.

Reprinted with permission from

Balfer, J.; Bajorath, J. Visualization and Interpretation of Support Vector Machine Activity Predictions. *J. Chem. Inf. Model.* **2015**, *55*, 1136–1147.

JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

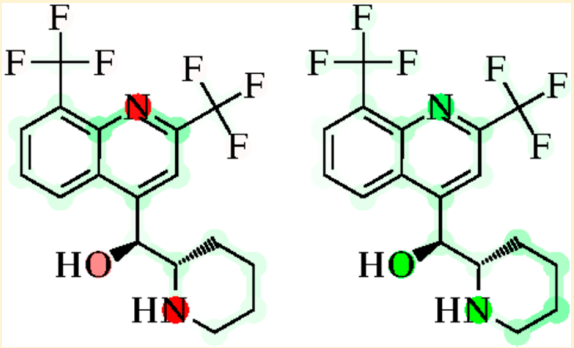
pubs.acs.org/jcim

Visualization and Interpretation of Support Vector Machine Activity Predictions

Jenny Balfer and Jürgen Bajorath*

Department of Life Science Informatics, B-IT, LIMES Program Unit Chemical Biology and Medicinal Chemistry, Rheinische Friedrich-Wilhelms-Universität, Dahlmannstr. 2, D-53113 Bonn, Germany

ABSTRACT: Support vector machines (SVMs) are among the preferred machine learning algorithms for virtual compound screening and activity prediction because of their frequently observed high performance levels. However, a well-known conundrum of SVMs (and other supervised learning methods) is the black box character of their predictions, which makes it difficult to understand why models succeed or fail. Herein we introduce an approach to rationalize the performance of SVM models based upon the Tanimoto kernel compared with the linear kernel. Model comparison and interpretation are facilitated by a visualization technique, making it possible to identify descriptor features that determine compound activity predictions. An implementation of the methodology has been made freely available.



INTRODUCTION

Support vector machines (SVMs) are among the most widely used machine learning algorithms in chemoinformatics,[1] especially for compound activity prediction. SVMs were originally used for binary object classification (e.g., active vs inactive compounds),[2] but have also been adapted for multitarget predictions[3−6] and compound ranking.[7,8] The popularity of SVMs is due to their ability to reach higher performance levels than other prediction methods in many applications.[1] A foundation of the SVM approach is the use of kernel functions to project data sets into higher-dimensional space representations in which a linear separation of positive and negative training instances is feasible. For ligand-based virtual screening, the Tanimoto kernel[9] is often used in combination with binary fingerprints as molecular representations. The Tanimoto kernel utilizes the well-known Tanimoto similarity formalism[10] and is parameter-free, which renders it attractive for chemoinformatics applications.

A conundrum of the SVM approach (and also other machine learning methods, such as neural networks) is its black box character, which refers to the inability to rationalize why prediction models succeed or fail and interpret them in chemical terms. This also means that it is generally difficult to modify models or molecular representations for specific applications. Only a few attempts to rationalize SVM modeling and performance have been made to date. Previous work on SVM model interpretation has usually focused on linearly separable data,[11,12] thereby avoiding analysis in high-dimensional kernel-dependent reference spaces. In the presence of nonlinear data−property relationships, data were partitioned into several local Voronoi regions prior to SVM modeling, and local SVM models were separately built for each of these regions.[11] The weights of the support vectors from which the models were generated were then used to assess the importance of each chosen molecular descriptor.[13,14] Another approach to assess the importance of descriptors in nonlinear SVMs internally stores information during kernel calculation and then readjusts these weights with linear SVM coefficients.[15] This method is only applicable if feature importance information is available for the given kernel. In addition, partial derivatives of a kernel function were used to identify descriptors with the largest gradient components, which were hypothesized to be the most important for prediction.[16] In this case, only the derivatives of the kernel function need to be provided.

Different from model internal analysis, "rule extraction" from SVMs has also been attempted.[17] In this case, one tries to mimic the classification of an SVM model as closely as possible, without interpreting the model itself, to derive a set of rules approximating SVM classification. Hence, these approaches aim at an indirect assessment of SVM predictions. Rule extraction often suffers from the lack of clear rule definitions and is difficult to apply in high-dimensional reference spaces,[17] a hallmark of SVM modeling.

Following a different approach, Hansen et al.[18] have proposed a method for prediction visualization in which the most important support vectors are displayed together with their factors. This method has the principal advantage that it does not require any prior knowledge about the kernel function used. However, in this case it is not possible to explain the influence of single descriptors or features on SVM classification. For this purpose, "explanation vectors" representing local gradients of input descriptors are derived. Therefore, the SVM model must also be mimicked by another classifier such as

Received: March 31, 2015
Published: May 19, 2015

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

Parzen windows.[19] The utility of such explanation vectors typically depends on the data sets under study.

Herein we introduce a new methodology for visualization and interpretation of SVM predictions using the Tanimoto kernel in comparison with the linear kernel. It provides intuitive access to descriptor features that are the most important for a given SVM prediction and enables mapping of features onto molecular graphs. An interactive graphical user interface is employed for visualization. The methodology clearly reveals how the Tanimoto kernel facilitates many accurate activity predictions and rationalizes failures of the linear kernel.

CONCEPTS AND METHODS

Support Vector Machine Theory. SVMs aim to solve a classification task by finding a hyperplane in feature space that best separates training examples having different binary class labels.[20] Test instances are then classified on the basis of the side of the separating hyperplane on which they fall, as determined by the following decision function:

$$f(\mathbf{x}) = \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle - b) \tag{1}$$

where $\mathbf{w}$ is the normal vector of the separating hyperplane, $\mathbf{x}$ is the test instance, and b is the so-called bias of the hyperplane. If $\{(\mathbf{x}^{(i)}, y^{(i)}) \mid i = 1, ..., n\}$ is a set of n training examples $\mathbf{x}^{(i)}$ with known class labels $y^{(i)} \in \{-1, +1\}$, the parameters $\mathbf{w}$ and b of the hyperplane are derived by solving the following optimization problem:[20]

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi^{(i)} \tag{2}$$

subject to

$$y^{(i)}(\langle \mathbf{w}, \mathbf{x}^{(i)} \rangle - b) \geq 1 - \xi^{(i)}, \quad i \in \{1, ..., n\}$$
$$\xi^{(i)} \geq 0, \quad i \in \{1, ..., n\} \tag{3}$$

Minimizing $\mathbf{w}$ yields the hyperplane with the maximum distance to training examples on either side, the so-called margin. The parameter C needs to be adjusted to control the balance between correct classification of training examples and permitted prediction errors, which is of critical importance for model generalization. Misclassifications are represented by the slack variables $\xi^{(i)}$ introduced to allow a certain number of training errors.[21]

Instead of directly solving the primal optimization problem, it is also possible to formulate an equivalent dual problem using Lagrangian multipliers.[20] In this formulation, the constraints of the original problem are added to the objective function, and the resulting dual problem must be optimized. By application of the Karush−Kuhn−Tucker conditions,[22] it is possible to formulate the dual problem for the primal optimization problem in eq 2 as follows:

$$\max\left(\sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle \right) \tag{4}$$

subject to

$$\sum_{i=1}^{n} \lambda^{(i)} y^{(i)} = 0$$
$$0 \leq \lambda^{(i)} \leq C, \quad i \in \{1, ..., n\} \tag{5}$$

where $\lambda^{(i)}$ are the Lagrangian multipliers that are introduced when the constraints of the primal problem are embedded into the objective function of the dual problem. This formulation makes it possible to compute the normal vector of the hyperplane as

$$\mathbf{w} = \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} \tag{6}$$

Since the Lagrangian multipliers $\lambda^{(i)}$ can be nonzero only for training examples that lie on or in the margin of the hyperplane or are misclassified, it is also possible to reduce eq 6 to this subset of training examples, the so-called support vectors.[20] Hence, the majority of training examples can be discarded following the training phase, which makes SVM modeling suitable for large data sets.

Another advantage of the dual formulation is that it enables the application of the "kernel trick".[23] The underlying idea is that data that cannot be linearly separated in the original feature space are projected into a higher-dimensional kernel space in which linear separation might become feasible. In this case, the normal vector $\mathbf{w}$ has the higher dimensionality of the kernel space, and training examples are projected into this new space via a mapping function $\phi(\mathbf{x})$. This changes only the constraints in eq 3:

$$y^{(i)}(\langle \mathbf{w}, \phi(\mathbf{x}^{(i)}) \rangle - b) \geq 1 - \xi^{(i)}, \quad i \in \{1, ..., n\} \tag{7}$$

Analogously, the dot product of the examples in eq 4 is replaced by the dot product of their mappings:

$$\max\left(\sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} \langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle \right) \tag{8}$$

Using Mercer's theorem,[24] we can replace the dot product in the dual objective function by a kernel function $K(u, v)$ that implicitly computes $\langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle$ without explicitly mapping $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ into the high-dimensional kernel space. To compute the normal vector of the hyperplane in kernel space, one would need to apply the explicit mapping $\phi(\mathbf{x})$:

$$\mathbf{w} = \sum_{\text{support vectors}} \lambda^{(i)} y^{(i)} \phi(\mathbf{x}^{(i)}) \tag{9}$$

In practice, this explicit derivation of $\mathbf{w}$ is not required because the decision function can also be expressed using the kernel:

$$f(x) = \mathrm{sign}(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle - b)$$
$$= \mathrm{sign}\left(\sum_{\text{support vectors}} \lambda^{(i)} y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) - b \right) \tag{10}$$

This procedure enables the use of kernel spaces that are theoretically infinite because the mapping functions do not need to be computed explicitly. A variety of kernel functions have been developed, including the Gaussian or radial basis function kernel, the Tanimoto kernel, and more complex graph kernels.[9,25,26] SVMs utilizing kernels usually have much higher prediction capacity than linear models.[1] However, the use of kernel functions comes at the price of black box character and lack of model interpretability. If the explicit mapping $\phi(\mathbf{x})$ is not available, it is impossible to determine contributions of the features to the classification.

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

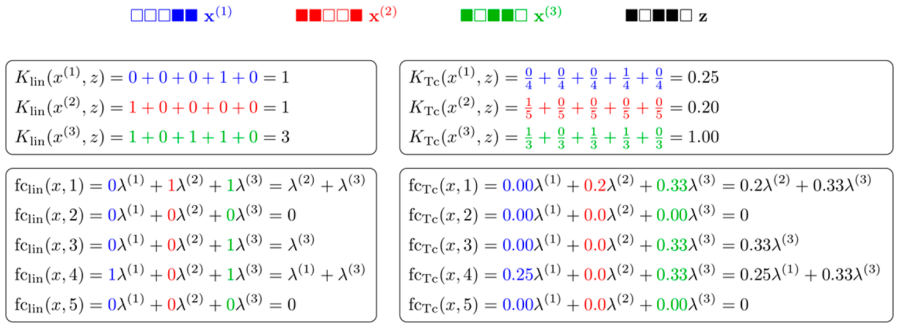


Figure 1. Example calculation. Shown is a minimal example consisting of three support vectors $\mathbf{x}^{(1)}$–$\mathbf{x}^{(3)}$ and one test compound $\mathbf{z}$, represented as fingerprints with five features. Filled and unfilled squares represent features that are set on and off, respectively. The support vectors are colored according to their contributions to the formulas shown below. On the left and right, example kernel and feature contribution calculations are shown for the linear and Tanimoto kernels, respectively.

Feature Weighting. For the linear kernel, feature importance can be easily interpreted because it can be expressed as a sum of individual feature contributions:

$$K_{\text{linear}}(\mathbf{u}, \mathbf{v}) = \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{d=1}^{D} u_d v_d \tag{11}$$

Thus, it is readily possible to weight each feature by $\lambda^{(i)} y^{(i)}$ and calculate the classification function as a sum of weighted feature contributions. However, nonlinear kernels often cannot be expressed as a sum of feature contributions. Nevertheless, they might be modified accordingly. For example, let us consider the Tanimoto kernel, defined as

$$\begin{aligned} K_{\text{Tanimoto}}(\mathbf{u}, \mathbf{v}) &= \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\langle \mathbf{u}, \mathbf{u} \rangle + \langle \mathbf{v}, \mathbf{v} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle} \\ &= \frac{\sum_{d=1}^{D} u_d v_d}{\langle \mathbf{u}, \mathbf{u} \rangle + \langle \mathbf{v}, \mathbf{v} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle} \\ &= \sum_{d=1}^{D} \frac{u_d v_d}{\langle \mathbf{u}, \mathbf{u} \rangle + \langle \mathbf{v}, \mathbf{v} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle} \end{aligned} \tag{12}$$

Under the condition that the denominator is constant, it is possible to express the Tanimoto kernel as a sum of feature contributions. Since $\mathbf{u}$ and $\mathbf{v}$ are constant for any single kernel calculation, the condition of a constant denominator is applicable in this case. To calculate $\text{fc}(\mathbf{x}, d)$, the contribution of feature d to an individual SVM prediction, the following equations are applied:

$$\text{fc}_{\text{linear}}(\mathbf{x}, d) = \sum_{\text{support vectors}} y^{(i)} \lambda^{(i)} x_d^{(i)} x_d \tag{13}$$

$$\text{fc}_{\text{Tanimoto}}(\mathbf{x}, d) = \sum_{\text{support vectors}} \frac{y^{(i)} \lambda^{(i)} x_d^{(i)} x_d}{\langle \mathbf{x}^{(i)}, \mathbf{x}^{(i)} \rangle + \langle \mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{x}^{(i)}, \mathbf{x} \rangle} \tag{14}$$

The denominator in eq 14 is constant only for each individual support vector. Nonetheless, it is possible to express the feature contribution as a sum. To clarify this point, we consider an exemplary case with three support vectors and five features, as shown in Figure 1. Here the three support vectors are labeled as $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$ and colored blue, red, and green, respectively, while the fingerprint of the compound to be predicted is shown in black and labeled as $\mathbf{z}$. For the linear kernel (left), the derivation is straightforward. The first two support vectors share one bit with the test compound and the third shares three. Accordingly, the feature contributions are derived (lower left). Here only $\lambda^{(i)} x_d^{(i)} x_d$ is shown, and the $y^{(i)}$ have been omitted for clarity. Since the second and fifth bits in the model fingerprint make no contribution to the kernel values, their feature contributions are zero. By contrast, the contributions of the first, third, and fourth bits are derived from the support vectors making a nonzero contribution to the respective sum.

On the right in Figure 1, the same calculations are reported for the Tanimoto kernel. Here the similarities of the support vectors and the test compound are weighted according to eq 12 and are therefore not the same for the first two support vectors. The denominator of each single kernel calculation is constant, but there are different denominators for each support vector. On the lower right, the derivations of the feature contributions are shown. Again, the second and fifth features do not contribute to the final prediction. However, the other three contributions are derived as weighted sums from the support vectors, in which not only $\lambda^{(i)}$ but also the different denominators contribute to the weighting. In the example shown, $\lambda^{(3)}$ has consistently higher weights than $\lambda^{(1)}$ or $\lambda^{(2)}$.

Prediction Visualization. To visualize SVM predictions, we use a graphical method reminiscent of the user interface previously introduced for visualization of naïve Bayesian classification models.[27] Each descriptor feature is visualized as a single point in a polar coordinate system. The more a feature contributes to the prediction, the more remote it is from the pole. Feature points making negative and positive contributions to a prediction are colored red and blue, respectively. Features can be organized into structural subsets displayed in different regions of the polar coordinate system. In the prototypical implementation we provide (see below), feature points can be interactively selected to access associated information. Figure 2 shows an exemplary visualization of a theoretical prediction wherein only three features make nonzero contributions to the prediction. These features include the substructure "NC(O)O" with a contribution of −0.1, the pattern "S=A" (where A refers to any aliphatic atom) with a contribution of 0.3, and the feature "halogen" with a contribution of 0.4. Hence, the final sum of these contributions is 0.6, meaning that this compound would be predicted as active in any model with a bias lower than 0.6. The bias, as defined above, can also be rationalized as a model threshold for the prediction of activity (i.e., if the sum of feature contributions exceeds the bias, a compound is predicted to be active). The relative percentage scale in Figure

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

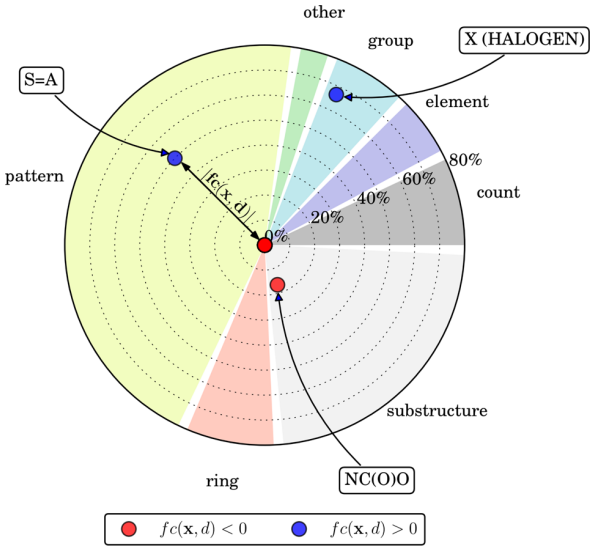


Figure 2. Principles of prediction visualization. Features are shown as points on a polar coordinate system and color-coded by positive (blue) and negative (red) feature contributions. The distance of a feature point from the pole reflects the magnitude of its contribution. Feature points are organized into groups and shown on differently colored backgrounds. A relative scale is provided that gives the percentage contribution of the feature to the overall sum.

2 can be used to easily access the relative importance of each feature. For instance, the halogen feature accounts for almost 67% of the final sum. While the distance of each point from the pole represents the magnitude of a feature contribution, colors refer to positive (blue) or negative (red) absolute contributions.

Feature Mapping. In addition to visualization of SVM predictions, feature contributions are also mapped back onto the molecular graph of the classified compound. For this purpose, we use an approach similar to that of Rosenbaum et al.[12] Each atom and bond in the molecular graph is assigned a weight accounting for its accumulated feature contributions. In the case of fingerprint descriptors, we first determine each feature that is set on and then locate the corresponding substructures in compound **x**. Each participating atom a_x and bond b_x is then assigned a feature contribution, and the contributions of overlapping features are added:

$$w(b_x) = \sum_{\{d|b_x \in d\}} \mathrm{fc}(\mathbf{x}, d) \tag{15}$$

For mapping, feature contributions are normalized with respect to the numbers of atoms and bonds in the corresponding substructures:

$$w(b_x) = \sum_{\{d|b_x \in d\}} \frac{n(d)\mathrm{fc}(\mathbf{x}, d)}{n_{\mathrm{atoms}}(\mathbf{x}, d) + n_{\mathrm{bonds}}(\mathbf{x}, d)} \tag{16}$$

where $n(d)$ is the number of times a substructure must occur in the molecule for feature d to be set. Usually, one occurrence of a substructure is required for a bit to be set on, but there are also features requiring more than one occurrence, such as "more than two nitrogens", for which $n(d) = 3$. This normalization is applied to ensure that atoms and bonds in large or recurrent substructures are assigned only a fraction of the feature weight while single-atom features are fully taken into account in the mapping.

The resulting weights are color-coded such that white corresponds to a weight of zero, red to a negative weight, and green to a positive weight. A weight of zero may occur if a certain atom or bond is not part of any substructure feature encoded by a fingerprint or if the contributions of overlapping features add up to zero. The color shading scales with the magnitude of negative or positive contributions (i.e., the darker the shading, the larger the magnitude).

MATERIALS AND PROTOCOLS

Compound Data Sets. From ChEMBL version 20,[28] three large sets of compounds active against different G-protein-coupled receptors with available high-confidence activity data for individual human targets were extracted, as summarized in Table 1. Compounds were required to be tested in direct binding assays with K_i values of at most 10 000 nM. In-house filters were applied to remove duplicate, highly reactive, and PAINS[29] compounds. Additionally, a set of mitogen-activated protein (MAP) kinase p38 alpha inhibitors was selected using the same criteria as above (Table 1). However, in this case, only IC_{50} measurements were available. Furthermore, 10 000 compounds not contained in these three data sets were randomly extracted from each of ChEMBL version 20 and ZINC version 12[30] and used as negative training examples.

Table 1. Data Sets[a]

TID	target name	no. of compounds
252	adenosine A2a receptor	2646
259	cannabinoid CB2 receptor	2202
72	dopamine D2 receptor	2200
10188	MAP kinase p38 alpha	1485

[a]For each data set, the ChEMBL target ID (TID), the target name, and the number of active compounds is reported.

Molecular Representation. In this study, we used the MACCS substructural fingerprint as a molecular representation.[31] The MACCS fingerprint consists of 166 bits, each of which encodes a predefined substructure or pattern. Fingerprint representations were computed with an in-house implementation using OpenEye's OEChem toolkit[32] and SMARTS patterns adapted from RDKit.[33] For visualization, MACCS features were organized into seven different groups, including "ring", "count", "group", "element", "substructure", "pattern", and "other".[27] This approach is applicable to any fingerprint.

Parameter Selection. First, we divided each data set into a training set containing 80% of its compounds and a test set with the remaining 20%. To select the best regularization term C for SVM models, 10-fold cross-validation was performed on the training set. For this purpose, the training set was randomly divided into 10 equally sized subsets. Each of these subsets was used once as a validation set, and models were built with varying C parameter on the remaining nine subsets. To assess the performance of all intermediate and final models, we used the F_1 score, defined as

$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{17}$$

where TP refers to true positive, FP to false positive, and FN to false negative predictions. Following parameter variation, the value of C yielding the best mean F_1 score across all subsets was selected.

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

To account for typically unbalanced SVM classification tasks (i.e., with more inactive than active compounds available), two adjusted versions of C, denoted as C_+ and C_-, can be used to account for the slack variables of positive and negative training examples, respectively:[34]

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C_+ \sum_{\{i|y^{(i)}=+1\}} \xi^{(i)} + C_- \sum_{\{i|y^{(i)}=-1\}} \xi^{(i)} \quad (18)$$

The terms C_+ and C_- are often derived such that their ratio is equal to the inverse ratio of active and inactive compounds:[34]

$$\frac{C_+}{C_-} = \frac{\text{no. of negative examples}}{\text{no. of positive examples}} \quad (19)$$

During cross-validation, C_- was first varied on a coarse grid of $\{2^x \mid x \in \{-5, -3, ...,15\}\}$ to preselect the best values, which was followed by cross-validation on a finer grid around preselected values of $\{2^x \mid x \in \{C_{best} - 2, C_{best} - 1.75, ..., C_{best} + 2\}\}$, leading to the final selection of C_-.[35] For each value of C_-, two models were trained: one with $C_+ = C_-$ and the other with a C_+ value derived via eq 19. The value combinations giving the best F_1 scores are summarized in Table 2. With one exception, the $C_+ = C_-$ setting was preferred, indicating that a potential influence of data imbalance was mostly negligible here.

Table 2. Model Parameters[a]

TID	kernel	C_-	C_+
252	linear	8.00	8.00
252	Tanimoto	45.25	45.25
259	linear	45.25	45.25
259	Tanimoto	26.91	128.74
72	linear	64.00	64.00
72	Tanimoto	32.00	32.00
10188	linear	0.11	0.11
10188	Tanimoto	64.00	64.00

[a]Given are the best values of C_- and C_+ for each compound set and kernel function as determined via cross-validation (see the text).

Final SVM Models. For each combination of activity class and kernel, the C_+ and C_- values reported in Table 2 were then used to train the final SVM prediction models on 80% of the compounds randomly selected from each activity class. Final model performance was assessed on the basis of F_1 scores derived on the test sets. All of the models were generated using the freely available implementation SVM^{light}.[36]

Software Used and Implementation. For feature mapping and compound display, OpenEye's OEChem and OEDepict toolkits[32,37] were used. Visualizations of predictions were generated using Matplotlib.[38] A prototypical Python implementation of the visualization methodology reported herein was made freely available via the open access platform Zenodo.[39]

RESULTS AND DISCUSSION

Interpretability of machine learning models is of high value for structure−activity relationship analysis.[12,16,18,27,40,41] For this purpose, we introduce an approach for the visualization of individual SVM predictions and identification of key features that determine activity predictions. Initially, as a basis for our investigation, we analyze SVM model performance for exemplary data sets and compare kernels. Then we focus on visualization, feature identification, and mapping.

Model Performance. Figure 3 summarizes the performance of the final SVM models using the ChEMBL subset as

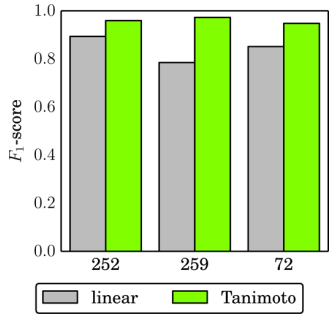


Figure 3. SVM model performance. Reported are the F_1 scores of the final models for the four target sets.

inactive training compounds. Generally high prediction accuracy was observed, with F_1 scores ranging from 78.5% to 97.3%. As anticipated, the use of the Tanimoto kernel led to more accurate prediction than the simple linear kernel, with differences in F_1 scores ranging from 6.6% (adenosine A2a receptor ligands) to 18.8% (cannabinoid CB2 receptor ligands). The models derived using ZINC compounds as inactive training examples yielded overall similar performance, with deviations below 2%. The only exception was the linear model for the cannabinoid CB2 receptor, which performed 7% better using ZINC compounds as inactives than ChEMBL compounds. Hence, unless stated otherwise, we will focus on the models derived using ChEMBL compounds as inactives in the following discussion.

Kernel Comparison. Tables 3−6 report predictions for the four activity classes to compare the two kernels at the level of

Table 3. Predictions for Adenosine A2a Receptor Ligands[a]

		linear			
		TP	FN	TN	FP
Tanimoto	TP	469	38		
	FN	6	8		
	TN			1935	45
	FP			7	22

[a]For the final SVM models, the numbers of true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs) are reported. The results are presented in a matrix format. For example, there were 469 TPs and 1935 TNs shared by SVM models using the Tanimoto and linear kernels, whereas six FNs using the Tanimoto kernel were TPs using the linear kernel and 45 TNs using the Tanimoto kernel were FPs using the linear kernel.

individual compounds, distinguishing between true positives (TPs), false negatives (FNs), true negatives (TNs), and false positives (FPs). Consistent with the overall high prediction accuracy, most of the compounds were correctly predicted to be active or inactive using both kernels (and are thus reported on the diagonal of the tables). However, for SVM model diagnostics and visualization, compounds yielding different predictions with alternative kernels (represented by off-diagonal numbers in tables) are prime examples.

From the subsets for which the Tanimoto models yielded correct predictions and the linear model incorrect predictions,

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

Table 4. Predictions for Cannabinoid CB2 Receptor Ligands[a]

		linear			
		TP	FN	TN	FP
Tanimoto	TP	318	89		
	FN	5	10		
	TN			1937	74
	FP			4	4

[a]For the final SVM models, the numbers of true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs) are reported. The data representation is according to Table 3.

Table 5. Predictions for Dopamine D2 Receptor Ligands[a]

		linear			
		TP	FN	TN	FP
Tanimoto	TP	384	45		
	FN	3	16		
	TN			1911	53
	FP			7	21

[a]For the final SVM models, the numbers of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are reported. The data representation is according to Table 3.

Table 6. Predictions for MAP Kinase p38 Alpha Inhibitors[a]

		linear			
		TP	FN	TN	FP
Tanimoto	TP	239	37		
	FN	1	12		
	TN			1962	32
	FP			5	10

[a]For the final SVM models, the numbers of true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs) are reported. The data representation is according to Table 3.

test compounds were selected having the largest difference of the predictive function:

$$\arg\max_{\mathbf{x}} |f_{\text{linear}}(\mathbf{x}) - f_{\text{Tanimoto}}(\mathbf{x})| \quad (20)$$

These compounds are given in Table 7, and their predictions are analyzed in the following. Table 7 also reports for each compound the sum of feature contributions from each prediction as well the SVM model biases for prediction of activity determined during the training phase. Cumulative feature contributions can be negative or positive. With increasing positive magnitude, the likelihood of positive activity predictions increases. A compound is predicted to be active if the sum of the feature contributions exceeds the model bias.

Table 7. Details of Individual Predictions[a]

		linear		Tanimoto	
TID	CID	$\sum$fc($\mathbf{x}$, d)	b	$\sum$fc($\mathbf{x}$, d)	b
252	CHEMBL411685	5.16	9.15	4.46	3.46
252	CHEMBL11113	12.94	9.15	3.15	3.46
259	CHEMBL1834525	−4.32	2.17	3.45	2.81
259	CHEMBL585041	5.10	2.17	2.15	2.81
72	CHEMBL419792	1.51	5.90	2.89	2.83
72	CHEMBL12028	8.14	5.90	2.12	2.83
10188	CHEMBL320069	2.08	5.74	4.33	3.33
10188	CHEMBL57	7.35	5.74	2.84	3.33

[a]For individual predictions discussed in the text, the ChEMBL target ID (TID), compound ID (CID), sum of feature contributions $\sum$fc($\mathbf{x}$, d), and model bias b for each kernel are reported. If the sum of the feature contributions is larger than the bias, the compound is predicted to be active; otherwise, it is predicted to be inactive.

Visualization of Predictions and Feature Mapping. The graphical analysis of predictions aims to identify descriptor features that make important contributions to correct and incorrect SVM predictions of compound activity using different kernel functions.

Adenosine A2a Receptor Ligands. Figure 4 shows prediction visualizations using the linear and Tanimoto kernels and mappings of MACCS fingerprint features for an active adenosine A2a receptor ligand. The compound was incorrectly predicted to be inactive by the linear kernel but correctly predicted to be active by the Tanimoto kernel. The visualization for the linear model (Figure 4 top) identified one feature with a large positive contribution to the prediction ("six-membered ring"), two features with intermediate positive contributions ("aromatic ring >1" and "C:N", where ":" denotes an aromatic bond), and one feature with an intermediate negative contribution ("N heterocycle"). Despite the preva-

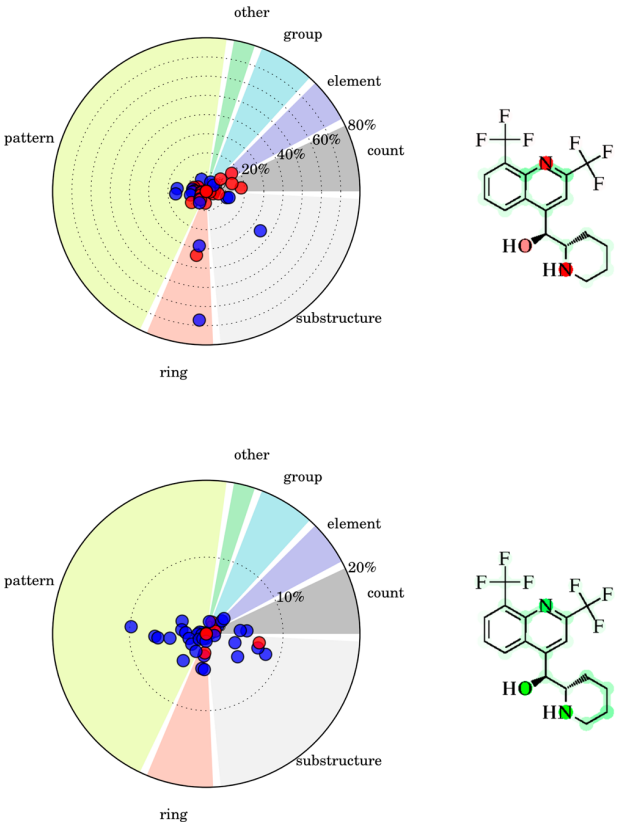


Figure 4. Visualization of predictions and feature mapping for ChEMBL compound 411685. Individual predictions using SVM models with the (top) linear and (bottom) Tanimoto kernels are visualized, and key structural features are mapped. In each panel showing predictions for a given compound, the relative scale of the visualizations has been adjusted such that the magnitudes of feature contributions can be directly compared. This adenosine A2a receptor ligand was predicted to be inactive by the linear and active by the Tanimoto model.

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

lence of three features making large or intermediate positive contributions, the linear model yielded a false negative prediction. All of the other MACCS features mapped to the center of the graph, corresponding to small-magnitude contributions. Although the six-membered ring made a feature contribution of almost 70% to the overall sum, the feature mapping shown in Figure 4 revealed that negative contributions of smaller magnitude also occurred at several ring atom positions, which reduced the positive contribution of the six-membered ring. Ultimately, the individual contributions led to a sum of feature contributions of 5.16, which was smaller than the bias of 9.15 for the linear model (Table 7). Therefore, the compound was predicted to be inactive. By contrast, the visualization of the correct prediction by the Tanimoto model (Figure 4 bottom) provides a different picture. All of the features made contributions of small magnitudes below 10% to the overall sum, and most of these contributions were positive. The feature mapping also showed that there was no single atom or bond in the molecule with a negative cumulative weight. In this case, the sum of feature contributions (4.46) clearly exceeded the relatively low bias of the Tanimoto model (3.46; Table 7), leading to a correct prediction of activity.

Figure 5 shows the results for an inactive test compound from the adenosine A2a receptor modeling, which was predicted to be active by the linear model. The visualization for the linear model (Figure 5 top) also identified the six-membered ring as a single dominant positive feature, although in this case it only accounted for less than 30% of the overall sum. Furthermore, there were a number of additional features with relatively small positive or negative contributions between 10% and 20%. Feature mapping identified a number of features with negative contributions centered at nitrogen and oxygen atoms throughout the molecule, similar to observations made for the linear model in Figure 4. However, in this case, the contribution of the six-membered ring and a part of the adjacent thiophene ring involving the sulfur atom clearly dominated the prediction, leading to a sum of 12.94 and a false positive assignment. The Tanimoto model correctly predicted this compound to be inactive. The visualization of this prediction (Figure 5 bottom) reveals the presence of many positive and negative contributions, especially from patterns. However, these contributions are of relatively small magnitude. As a result of the presence of a variety of positive and negative feature contributions in the Tanimoto model, which partly compensated for each other, the sum of contributions (3.15) for the compound in Figure 5 did not reach the model bias (3.46), leading to a correct prediction of inactivity. Furthermore, feature mapping showed that there were only few oxygen atoms with an overall negative weight.

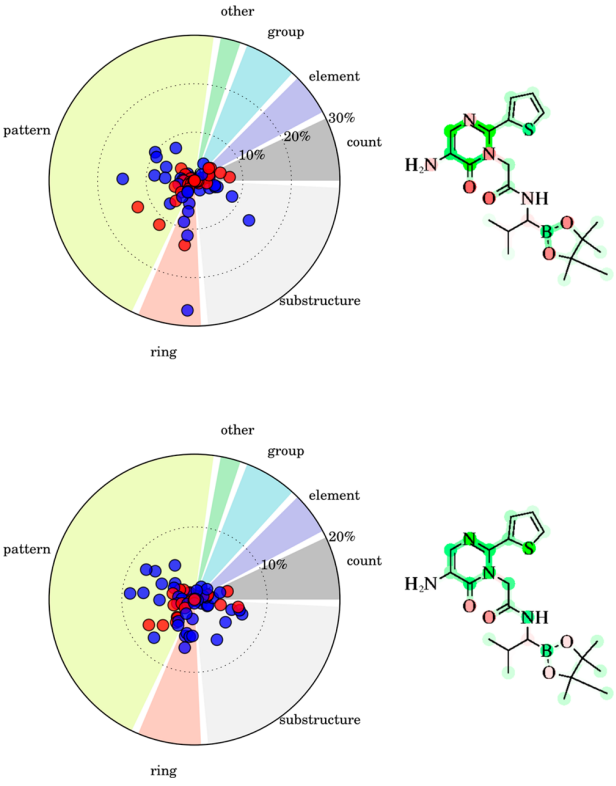


Figure 5. Visualization of predictions and feature mapping for ChEMBL compound 1113, represented according to Figure 4. This compound is a negative test example for prediction of adenosine A2a receptor ligands. Predictions: active (linear model) and inactive (Tanimoto model).

In general, features shared by predictions using the linear and Tanimoto kernels often made contributions of lesser magnitude to the Tanimoto model. This difference is a direct consequence of the denominator in the Tanimoto kernel, which weights the numbers of fingerprint bits set on for two compounds by the total number of possible commonly set bits. Thus, different from the linear kernel, the Tanimoto kernel further differentiates between compound pairs sharing a large number of bits but differing in their sizes and total numbers of bits set to 1.

Cannabinoid CB2 Receptor Ligands. Figure 6 shows an active cannabinoid CB2 receptor ligand that was predicted to be inactive by the linear model and active by the Tanimoto model. The visualization of the prediction using the linear model (Figure 6 top) highlights the dominance of a single feature (pattern "QSQ", where Q refers to any heteroatom) that makes a strong negative contribution of over 90% of the cumulative sum. In addition, a comparable number of other features with intermediate positive or negative contributions became apparent. Feature mapping revealed that the "QSQ" pattern was centered on an individual sulfur atom contained in this compound. However, the neighboring carbon and oxygen atoms made only minor or no negative contributions to the prediction, while the sulfur atom itself had a strong negative net effect. Furthermore, the nitrogen atom made a significant contribution to the negative prediction because it was a part of several features with moderately negative effects. This example illustrates the utility of the feature mapping to highlight focused contributions in cases where individual atoms participate in multiple features influencing a prediction. The prediction of this compound using the Tanimoto kernel is visualized at the bottom of Figure 6. In this case, many features with positive contributions of small magnitude were observed, but only few with negative contributions approaching the 10% limit, thus rationalizing the positive prediction. This was also consistent with the view obtained from feature mapping, which revealed that all of the atoms, including the nitrogen and sulfur, and all of the bonds had positive cumulative weights.

Figure 7 visualizes predictions for a negative test example subjected to cannabinoid CB2 receptor ligand modeling, which was predicted to be active by the linear model and inactive by the Tanimoto model. In this case, the prediction visualization using the linear kernel (Figure 7 top) shows different

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

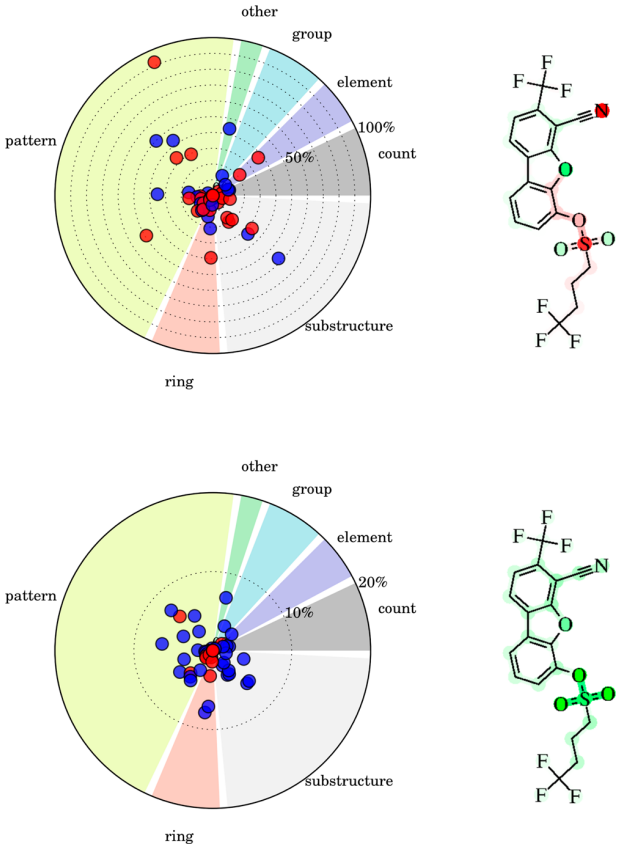


Figure 6. Visualization of predictions and feature mapping for ChEMBL compound 1834525, represented according to Figure 4. This compound is a cannabinoid CB2 receptor ligand. Predictions: inactive (linear), active (Tanimoto).

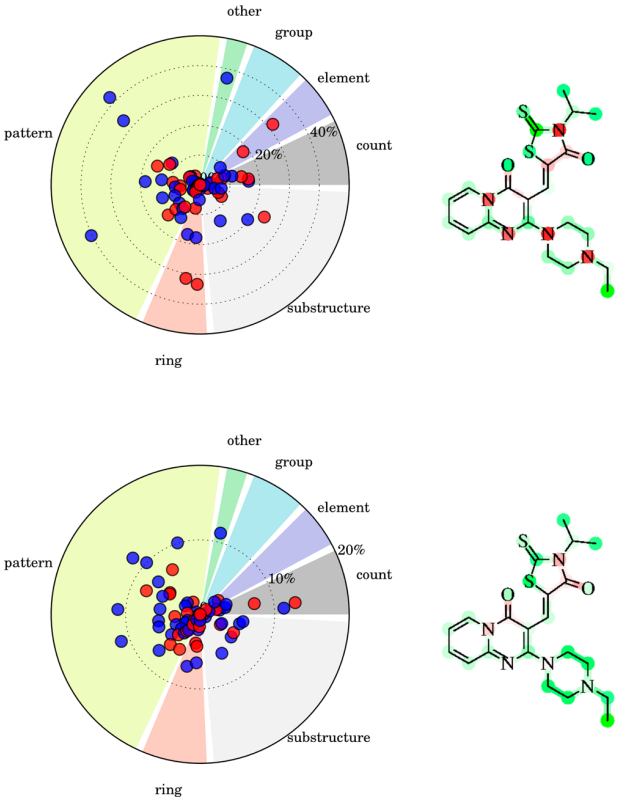


Figure 7. Visualization of predictions and feature mapping for ChEMBL compound 585041, represented according to Figure 4. This compound is a negative test example for prediction of cannabinoid CB2 receptor ligands. Predictions: active (linear), inactive (Tanimoto).

characteristics than the one discussed above. There were four features with strong positive contributions, three of which were patterns ("S=A", "A!N$A", and "NAO", where A refers to any aliphatic atom) and one was from the "other" group ("aromatic"). In addition, four features with negative contributions exceeding 30% of the overall sum included two rings ("N heterocycle", "ring"), a substructure ("OC(N)C"), and an element ("N"). Because two of the positive features and three of the negative features contained a nitrogen feature, mapping was crucial in this case to determine the contributions of these atoms. Mapping revealed that all of the nitrogen atoms ultimately made net contributions to the prediction of inactivity, whereas oxygen and sulfur atoms made contributions to the false prediction of activity, exceeding the bias of the linear model (Table 7). Interestingly, the feature mapping for the Tanimoto model (Figure 7 bottom) indicated that most of the atoms and patterns, except nitrogens and a few bond patterns, made positive contributions. This was also reflected in the prediction visualization, where the occurrence of a single nitrogen did not make any notable contribution; the only feature with a negative contribution exceeding the 10% limit was "A$A!O>1", which covered all of the nitrogen atoms in the compound. Hence, some of the nitrogens had a small cumulative negative weight, but overall there were many compensatory effects, and the sum of all contributions (2.15) was slightly smaller than the bias of the Tanimoto model (2.81; Table 7), leading to the correct prediction of inactivity. However, the numerical difference between the sum arising from multiple positive and negative feature contributions and the bias was relatively small in this case, consistent with the visualization in Figure 7.

Dopamine D2 Receptor Ligands. The active compound depicted in Figure 8 is small, consisting of only a condensed three-ring system. It was predicted to be inactive by the linear model and active by the Tanimoto model. The visualization of the prediction using the linear kernel (Figure 8 top) reveals a substructure feature ("CN(C)C") with a large positive contribution of more than 200% of the final sum. However, two patterns ("AN(A)A" and "AQ(A)A", where A refers to an aliphatic atom and Q to a heteroatom) with negative contributions of about 100% of the final sum nearly nullified this effect because the substructure CN(C)C matched both of these patterns. Aside from these features, the "aromatic" feature made the largest contribution. Feature mapping showed that cumulative negative contributions were mostly centered on the three nitrogen atoms, while positive contributions were distributed over the ring atoms (resulting in small atom-centric contributions). Overall, the negative and positive feature contributions were nearly compensatory, resulting in a cumulative feature contribution of 1.51, which was much smaller than the model bias of 5.90 (Table 7). The prediction using the Tanimoto model (Figure 8 bottom) was numerically vulnerable to boundary effects since the cumulative positive contribution of 2.89 only slightly exceeded the model bias of 2.83 (Table 7). However, the prediction visualization reveals a

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

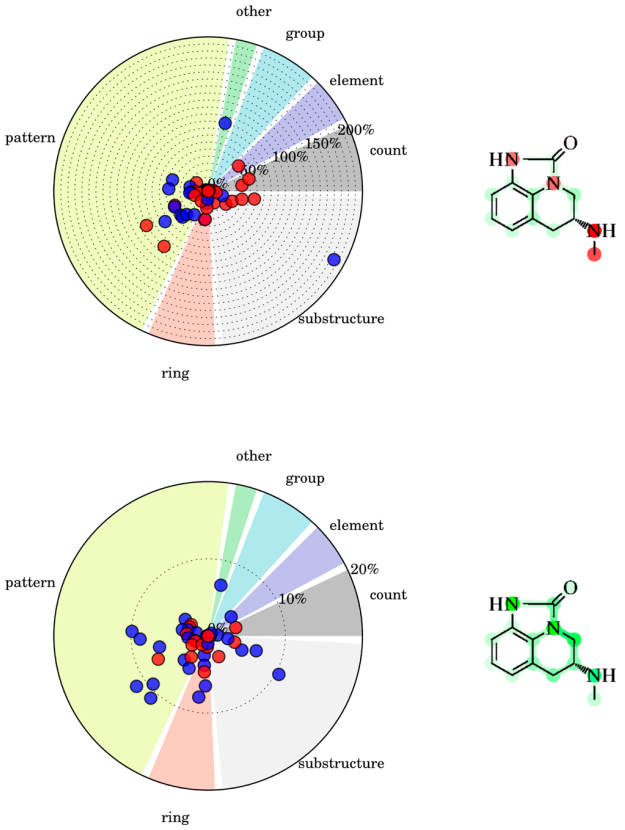


Figure 8. Visualization of predictions and feature mapping for ChEMBL compound 419792, represented according to Figure 4. This compound is a dopamine D2 receptor ligand. Predictions: inactive (linear), active (Tanimoto).

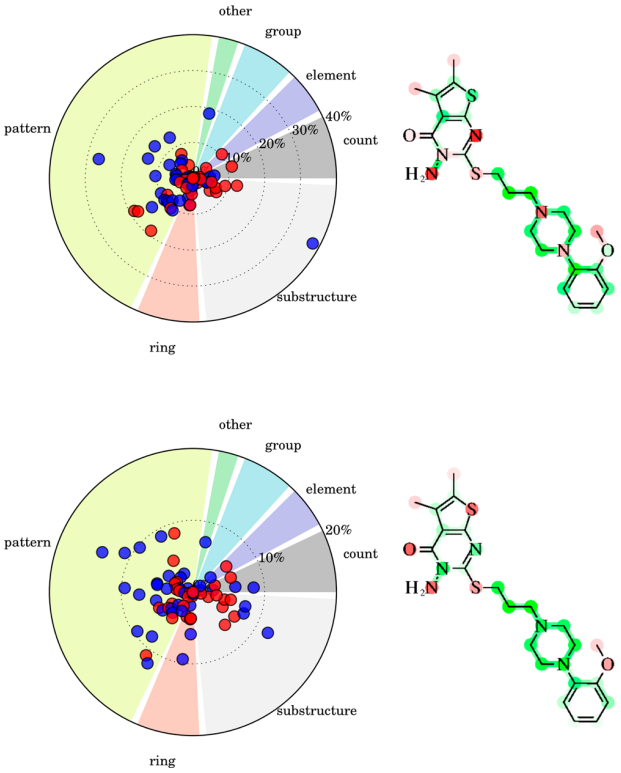


Figure 9. Visualization of predictions and feature mapping for ChEMBL compound 12028, represented according to Figure 4. This compound is a negative test example for prediction of dopamine D2 receptor ligands. Predictions: active (linear), inactive (Tanimoto).

variety of positive feature contributions (more so than negative ones), also including the "CN(C)C" substructure at the 10% level. However, all of the contributions were of low magnitude (with a maximum value of 0.33, corresponding to 11.34% of the overall sum). Nonetheless, despite the overall low cumulative feature contribution comparable to the model bias, feature mapping revealed only positive net contributions at atoms and bonds across the molecule.

Figure 9 visualizes predictions for a negative test example in dopamine D2 receptor ligand modeling that was predicted to be active by the linear model and inactive by the Tanimoto model. This molecule is much larger than the active compound in Figure 8. The visualization for the linear model (Figure 9 top) reveals the largest positive contributions to come from a substructure ("CN(C)C"), a pattern ("QN"), and another feature ("aromatic"), while negative contributions mostly originated from patterns ("AQ(A)A", "AN(A)A", and "QQ"). Again, contributions of these features largely compensated for each other because the substructure CN(C)C also matched AQ(A)A and AN(A)A and the same fragment matched patterns QN and QQ. Nonetheless, the sum of contributions from predictions using the linear kernel in Figure 9 was 8.14, which clearly exceeded the model bias of 5.90 (Table 7), leading to a false positive prediction. Feature mapping showed that negative contributions, mostly from nitrogen atoms, were too small to match cumulative positive contributions from the linker and ring systems of the compound. The visualization of the prediction using the Tanimoto kernel (Figure 9 bottom) provides a different picture. In this case, many features with positive or negative contributions of varying magnitudes were identified, and there were no individual features that largely determined the predictions. Feature mapping revealed the presence of multiple positive and negative contributions in corresponding regions of the negative test compound using the linear and Tanimoto kernels. However, in the case of the Tanimoto model, nitrogen atoms mostly made positive contributions, while oxygen and sulfur atoms made negative contributions, which was different from the linear model. Furthermore, the contributions of the aromatic ring systems to the prediction using the Tanimoto kernel were of lesser magnitude compared with the linear kernel. Overall, the Tanimoto kernel yielded a number of compensatory positive and negative feature contributions, and the final sum (2.12) did not reach the model bias of 2.83 (Table 7).

MAP Kinase p38 Alpha Inhibitors. Figure 10 shows an active MAP kinase p38 alpha inhibitor that was incorrectly predicted to be inactive by the linear kernel and correctly predicted to be active by the Tanimoto kernel. The prediction visualization revealed that there were many features in the linear prediction that contributed in a similar way. The strongest positive features were the substructure "NH", the ring feature "aromatic ring >1", and the pattern "NA(A)A". The features with strongest negative contributions included the substructure "C−N" and the pattern "NACH2A". Overall, the mapping showed that nitrogen and oxygen atoms made overall positive contributions, while the linker in the lower part of the molecule made the only considerable negative contribution. The overall sum of feature contributions was 2.08, which was

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

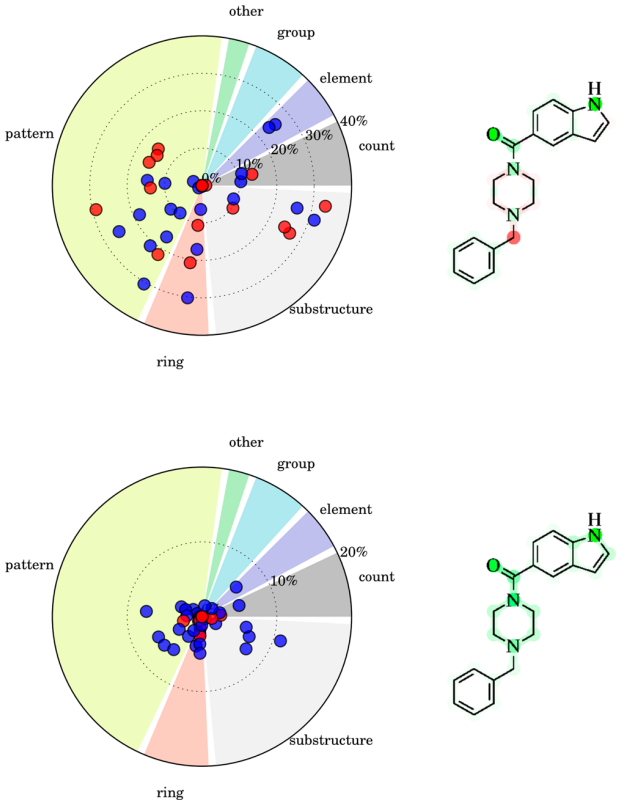


Figure 10. Visualization of predictions and feature mapping for ChEMBL compound 320069, represented according to Figure 4. This compound is a positive test example for prediction of MAP kinase p38 alpha ligands. Predictions: inactive (linear), active (Tanimoto).

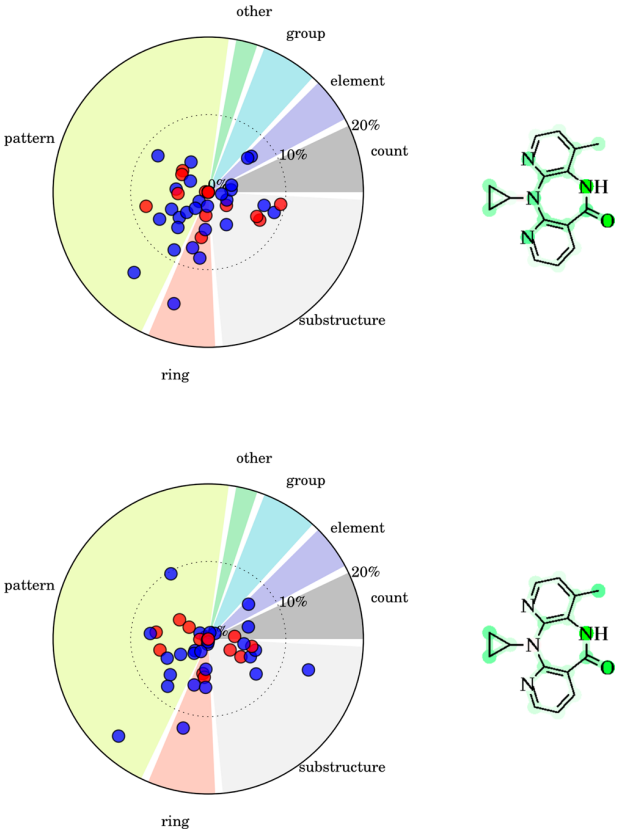


Figure 11. Visualization of predictions and feature mapping for ChEMBL compound 57, represented according to Figure 4. This compound is a negative test example for prediction of MAP kinase p38 alpha ligands. Predictions: active (linear), inactive (Tanimoto).

considerably smaller than the threshold of 5.74, meaning that the few cumulative positive contributions were not sufficient to yield a prediction of activity. By contrast, the visualization of the Tanimoto prediction revealed a different picture. Here only one substructure ("NH") having a considerable positive contribution was identified, in addition to many smaller positive contributions. There were very few if any negative contributions, as was also confirmed by feature mapping, which identified only cumulative positive contributions. Here the sum of the feature contributions of 4.33 exceeded the model bias by 1 (Table 7).

Furthermore, Figure 11 shows the prediction visualization and mappings for an inactive compound that was predicted to be active by the linear model and inactive by the Tanimoto model. The linear prediction visualization identified two important positive contributions, including a seven-membered ring and the pattern "A!:A:A!:A" (where ":" refers to an aromatic bond and "!" denotes negation). Many other features contributed only less than 10% to the overall sum. Here the prediction visualization showed that there were more features with positive contributions than negative ones. Features covering nearly the entire compound contributed to the positive prediction, leading to a large sum of 7.35 (Table 7). By contrast, the Tanimoto kernel's contributions yielded a sum of 2.84, which did not reach the model bias of 3.33, leading to a correct inactive prediction. In this case, three features contributed more than 10% to the overall sum: the pattern "A!:A:A!:A", the three-membered ring, and the substructure "NH". There were also several features with small positive or negative contributions. Feature mapping revealed that three nitrogen atoms made much smaller contributions compared with the linear case, leading to a smaller sum and thus the prediction of inactivity.

Taken together, the visualizations of the exemplary predictions in Figures 4−11 in combination with feature mapping helped to rationalize kernel-dependent differences in activity predictions. While several predictions using the simple linear kernel were dominated by individual feature contributions, the Tanimoto kernel — given its design, as discussed above — better differentiated feature contributions and their relative magnitudes. The Tanimoto kernel also generally reduced the magnitude of feature contributions, thereby balancing the influence of individual features on the predictions. Feature mapping complemented the visualization of predictions by focusing on atoms, bonds, or other substructures (e.g., rings) that were involved in multiple features and accounting for net effects. The exemplary predictions summarized in Table 7 also illustrate that cumulative feature contributions were rarely negative, even for compounds correctly predicted to be inactive — a previously unobserved effect. In these cases, the positive cumulative contributions were smaller than the model biases.

Variation of Inactive Training Compounds. All of the calculations discussed herein with negative training examples taken from ChEMBL were repeated with an equally sized negative training set randomly selected from ZINC. In order to analyze the effect of different inactive training sets on our visualization method, prediction visualization and feature

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

mappings of active compounds were compared. Figure 12 shows the comparison of the linear and Tanimoto models for

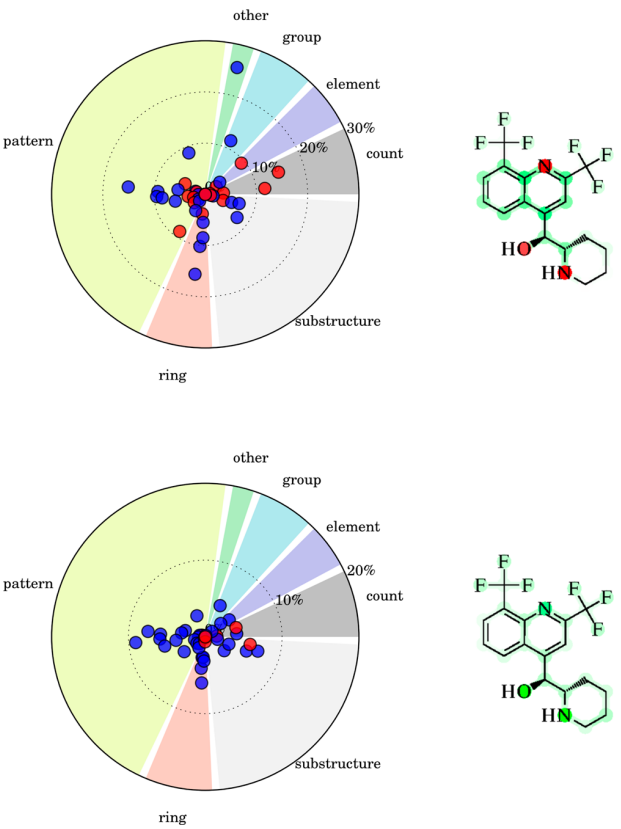


Figure 12. Visualization of predictions and feature mapping for the adenosine A2a receptor ligand from Figure 4 obtained using a subset of ZINC compounds as inactive training instances and the (top) linear and (bottom) Tanimoto models.

the adenosine A2a receptor ligand analyzed in Figure 4 using the ZINC training set. The compound was predicted to be inactive by the linear model and active by the Tanimoto model, regardless of the choice of the inactive training set. Feature mapping revealed that the same atoms and bonds contributed positively or negatively to the prediction in both cases. However, the prediction visualization showed that the features leading to these cumulative contributions differed. For instance, the most important positive and negative features for the linear ChEMBL-based models were the six-membered ring and the N heterocycle, respectively, as discussed above. By contrast, for the linear ZINC-based model, the "aromatic" feature from the "other" group was the most important positive feature and the "N>1" feature from the "count" group the most important negative feature. However, the Tanimoto models derived using ChEMBL and ZINC subsets did not differ notably.

Figure 13 shows the active cannabinoid CB2 receptor ligand from Figure 6 together with its prediction visualization and feature mappings for the linear and Tanimoto models based upon the inactive training set from ZINC. The compound was predicted to be inactive by the linear model and active by the Tanimoto model. However, both prediction visualization and feature mapping of the linear model differed from those shown in Figure 6. For example, the pattern "S=A" made by far the largest positive contribution in the linear ZINC model. This

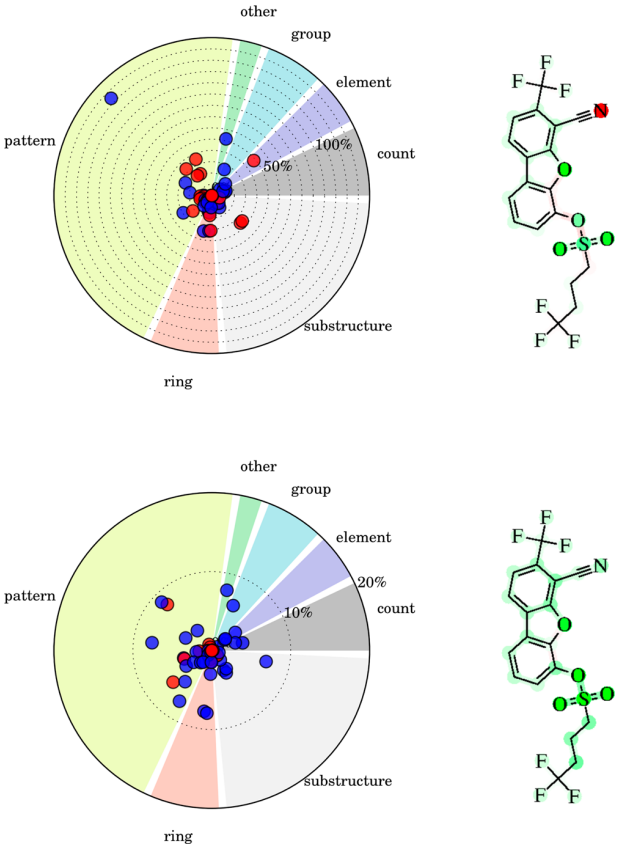


Figure 13. Visualization of predictions and feature mapping for the cannabinoid CB2 receptor ligand from Figure 6 obtained using a subset of ZINC compounds as inactive training instances and the (top) linear and (bottom) Tanimoto models.

contribution caused a change in the feature mapping from the sulfur atom making a negative contribution (ChEMBL-based model, Figure 6) to a positive contribution (ZINC-based model, Figure 13). Other features with contributions of ∼50% to the overall sum in the ChEMBL model made smaller contributions to the ZINC model. By contrast, the Tanimoto models using the ChEMBL and ZINC subsets displayed very similar characteristics in prediction visualization and feature mapping. The examples in Figures 12 and 13 show how the choice of negative training data might influence an SVM model and how prediction visualization and feature mapping can be used to analyze this influence.

CONCLUSIONS

In this study, we have introduced a visualization method for SVM predictions using the Tanimoto kernel compared with the linear kernel. Our analysis has revealed how activity predictions are determined by contributions from varying numbers of fingerprint features. The study can be extended to other kernel functions for which feature contributions can be expressed as sums. Visualization complemented by feature mapping provides a direct diagnostic for SVM models and reduces the black box character of SVM predictions. An implementation of the visualization approach introduced herein has been made freely available to aid in the assessment of SVM models and their successes and failures.

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

AUTHOR INFORMATION

Corresponding Author

*Tel: +49-228-2699-306. Fax: +49-228-2699-341. E-mail: bajorath@bit.uni-bonn.de.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The use of OpenEye's OEChem and OEDepict Toolkit was made possible by their free academic licensing program.

REFERENCES

(1) Heikamp, K.; Bajorath, J. Support Vector Machines for Drug Discovery. *Expert Opin. Drug Discovery* **2014**, *9*, 93−104.

(2) Geppert, H.; Humrich, J.; Stumpfe, D.; Gärtner, T.; Bajorath, J. Ligand Prediction from Protein Sequence and Small Molecule Information Using Support Vector Machines and Fingerprint Descriptors. *J. Chem. Inf. Model.* **2009**, *49*, 767−779.

(3) Erhan, D.; L'Heureux, P.-J.; Yue, S. Y.; Bengio, Y. Collaborative Filtering on a Family of Biological Targets. *J. Chem. Inf. Model.* **2006**, *46*, 626−635.

(4) Jacob, L.; Vert, J.-P. Protein−Ligand Interaction Prediction: An Improved Chemogenomics Approach. *Bioinformatics* **2008**, *24*, 2149−2156.

(5) Kawai, K.; Fujishima, S.; Takahashi, Y. Predictive Activity Profiling of Drugs by Topological-Fragment-Spectra-Based Support Vector Machines. *J. Chem. Inf. Model.* **2008**, *48*, 1152−1160.

(6) Heikamp, K.; Bajorath, J. Prediction of Compounds with Closely Related Activity Profiles Using Weighted Support Vector Machine Linear Combinations. *J. Chem. Inf. Model.* **2013**, *53*, 791−801.

(7) Wassermann, A. M.; Heikamp, K.; Bajorath, J. Potency-Directed Similarity Searching Using Support Vector Machines. *Chem. Biol. Drug Des.* **2011**, *77*, 30−38.

(8) Rathke, F.; Hansen, K.; Brefeld, U.; Müller, K.-R. StructRank: A New Approach for Ligand-Based Virtual Screening. *J. Chem. Inf. Model.* **2011**, *51*, 83−92.

(9) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Networks* **2005**, *18*, 1093−1110.

(10) Rogers, D. J.; Tanimoto, T. T. A Computer Program for Classifying Plants. *Science* **1960**, *132*, 1115−1118.

(11) Navia-Vázquez, A.; Parrado-Hernández, E. Support Vector Machine Interpretation. *Neurocomputing* **2006**, *69*, 1754−1759.

(12) Rosenbaum, L.; Hinselmann, G.; Jahn, A.; Zell, A. Interpreting Linear Support Vector Machine Models with Heat Map Molecule Coloring. *J. Cheminf.* **2011**, *3*, No. 11.

(13) Franke, L.; Byvatov, E.; Werz, O.; Steinhilber, D.; Schneider, P.; Schneider, G. Extraction and Visualization of Potential Pharmacophore Points Using Support Vector Machines: Application to Ligand-Based Virtual Screening for COX-2 Inhibitors. *J. Med. Chem.* **2005**, *48*, 6997−7004.

(14) Devos, O.; Ruckebusch, C.; Durand, A.; Duponchel, L.; Huvenne, J.-P. Support Vector Machines (SVM) in Near Infrared (NIR) Spectroscopy: Focus on Parameters Optimization and Model Interpretation. *Chemom. Intell. Lab. Syst.* **2009**, *96*, 27−33.

(15) Mohr, J.; Jain, B.; Sutter, A.; Laak, A. T.; Steger-Hartmann, T.; Heinrich, N.; Obermayer, K. A Maximum Common Subgraph Kernel Method for Predicting the Chromosome Aberration Test. *J. Chem. Inf. Model.* **2010**, *50*, 1821−1838.

(16) Carlsson, L.; Helgee, E. A.; Boyer, S. Interpretation of Nonlinear QSAR Models Applied to Ames Mutagenicity Data. *J. Chem. Inf. Model.* **2009**, *49*, 2551−2558.

(17) Martens, D.; Huysmans, J.; Setiono, R.; Vanthienen, J.; Baesens, B. Rule Extraction from Support Vector Machines: An Overview of Issues and Application in Credit Scoring. *Stud. Comput. Intell.* **2008**, *80*, 33−63.

(18) Hansen, K.; Baehrens, D.; Schroeter, T.; Rupp, M.; Müller, K.-R. Visual Interpretation of Kernel-Based Prediction Models. *Mol. Inf.* **2011**, *30*, 817−826.

(19) Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; Müller, K.-R. How To Explain Individual Classification Decisions. *J. Mach. Learn. Res.* **2010**, *11*, 1803−1831.

(20) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.

(21) Cortes, C.; Vapnik, V. N. Support Vector Networks. *Mach. Learn.* **1995**, *20*, 273−297.

(22) Kuhn, H. W.; Tucker, A. W. Nonlinear Programming. *Proc. Berkeley Symp. Math., Stat. Probab., 2nd* **1950**, 481−492.

(23) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A Training Algorithm for Optimal Margin Classifiers. *Proc. Annu. Workshop Comput. Learn. Theory, 5th* **1992**, 144−152.

(24) Mercer, J. Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philos. Trans. R. Soc. London, Ser. A* **1909**, *209*, 415−446.

(25) Gärtner, T.; Flach, P.; Wrobel, S. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Learning Theory and Kernel Machines*; Springer: Berlin, 2003.

(26) Kashima, H.; Tsuda, K.; Inokuchi, A. Marginalized Kernels between Labeled Graphs. *Proc. Int. Conf. Mach. Learn., 20th* **2003**, 321−328.

(27) Balfer, J.; Bajorath, J. Introduction of a Methodology for Graphical Interpretation of Naïve Bayesian Classification Models. *J. Chem. Inf. Model.* **2014**, *54*, 2451−2468.

(28) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Kruger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; Nowotka, M.; Papadatos, G.; Santos, R.; Overington, J. P. The ChEMBL Bioactivity Database: An Update. *Nucleic Acids Res.* **2014**, *42*, 1083−1090.

(29) Baell, J. B.; Holloway, G. A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *J. Med. Chem.* **2010**, *53*, 2719−2740.

(30) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool To Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.

(31) *MACCS Structural Keys*; Accelrys: San Diego, CA, 2011.

(32) *OEChem Toolkit*, version 2.0.2.; OpenEye Scientific Software: Santa Fe, NM; http://www.eyesopen.com.

(33) RDKit: Open-Source Cheminformatics. http://www.rdkit.org.

(34) Morik, K.; Brockhausen, P.; Joachims, T. Combining Statistical Learning with a Knowledge-Based Approach—A Case Study in Intensive Care Monitoring. *Proc. Int. Conf. Mach. Learn., 16th* **1999**, 268−277.

(35) Hsu, C.-W; Chang, C.-C.; Lin, C.-J. *A Practical Guide to Support Vector Classification*. Technical Report; Department of Computer Science, National Taiwan University: Taipei, Taiwan, 2003.

(36) Joachims, T. Making Large-Scale Support Vector Machine Learning Practical. In *Advances in Kernel Methods*; Schölkopf, B., Burges, C. J. C., Smola, A. J., Eds.; MIT Press: Cambridge, MA, 1999; pp 169−184.

(37) *OEDepict Toolkit*, version 2.2.4.; *OpenEye Scientific Software*: Santa Fe, NM; http://www.eyesopen.com.

(38) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90−95.

(39) Balfer, J.; Bajorath, J. Visualization and Interpretation of Support Vector Machine Activity Predictions. DOI: 10.5281/zenodo.17718.

(40) Marcou, G.; Horvath, D.; Solov'ev, V.; Arrault, A.; Vayer, P.; Varnek, A. Interpretability of SAR/QSAR Models of Any Complexity by Atomic Contributions. *Mol. Inf.* **2012**, *31*, 639−642.

(41) Polishchuk, P. G.; Kuz'min, V. E.; Artemenko, A. G.; Muratov, E. N. Universal Approach for Structural Interpretation of QSAR/QSPR Models. *Mol. Inf.* **2013**, *32*, 843−853.

DOI: 10.1021/acs.jcim.5b00175
J. Chem. Inf. Model. 2015, 55, 1136−1147

Summary

In this study, we have compared the linear and Tanimoto kernel in SVM modeling using molecular fingerprints. It was shown how individual compounds were predicted by the different kernels, and how individual features influenced the predictions. Furthermore, the cumulative feature contributions were rarely found to be negative. Instead, the SVM model bias, which was positive in all discussed cases, prevented false positive predictions.

A prototypic Python implementation of the method is freely available under the DOI `10.5281/zenodo.17718`.

This publication aims to extend the model and prediction visualization approach of the previous chapter to more complex models. We hope that our contribution motivates further research into the direction of intuitive model assessment. While the approach shown in this study is only applicable to SVMs using fingerprints and linear or Tanimoto kernels, similar methods could be developed for other models and kernel functions.

Conclusion

Computational drug discovery is an interdisciplinary field at the interface of a variety of disciplines, first and foremost medicinal chemistry and computer science. Today, many problems of the drug development process are attempted to be solved via *in silico* modeling. In part I of this thesis, two such methods were introduced and discussed.

Sometimes even more important than the successful application of machine learning models is the ability to understand and interpret them in chemical terms. Part II of this thesis provided important insights into *in silico* models for activity and potency prediction. Chapter 3 highlighted how chemogenomics data can be used for compound structure-independent activity prediction. This methodology allows the application of LBVS outside the applicability domain of the prominent similarity-property principle. In chapter 4, it was shown that even models that give globally satisfying results are not always reliable for compound potency prediction. Careful analysis revealed that they tend to predict continuous SARs and therefore miss the most potent compounds, which fall into discontinuous regions.

Eventually, part III introduced two visualization methods for fingerprint-based naïve Bayes classifiers and SVMs using the Tanimoto kernel. The main contribution here is the presentation of a method to analyze successful LBVS models in a way that is both formally precise and chemically interpretable. To our best knowledge, the visualization of the model and prediction itself is the first attempt in this direction. It complements the feature mapping onto the molecular graph, which is often used for model interpretation. We hope that these methods can contribute to an easier communication between the different domain experts involved in drug discovery.

Finally, we would like to stress potential opportunities for future research. Chapter 4 revealed systematic modeling artifacts in SVR for potency prediction, yet no solution to this issue has been offered thus far. One possibility would be the design of a kernel function that takes discontinuity information into account. Previously, it has been shown that it is possible to predict whether a pair of compounds form an activity cliff or not [91]. If this method was extended to predict the potency difference of a compound pair, this information could be incorporated into the final SVR kernel.

Another open subject is the extension of the methods provided in part III to new machine learning models or SVM kernels. Then, not only a quantitative comparison of model performances could be made, as is often done in classical LBVS benchmarking studies. Instead, one could pursue a more qualitative comparison of different models in terms of feature prioritization.

Bibliography

(1) Jones, A. W. Early Drug Discovery and the Rise of Pharmaceutical Chemistry. *Drug Test. Anal.* **2011**, *3*, 337–344.

(2) Mukherjee, S., *Der König aller Krankheiten. Krebs - eine Biografie*; Dumont: 2012.

(3) Mullard, A. New Drugs Cost US$2.6 Billion to Develop. *Nat. Rev. Drug Discov.* **2014**, *13*, 877.

(4) Tufts Center for the Study of Drug Development Cost to Develop and Win Marketing Approval for a New Drug is $2.6 Billion., `http://csdd.tufts.edu/news/complete_story/pr_tufts_csdd_2014_cost_study`, accessed April 2015.

(5) Lundstrom, K., An Overview on GPCRs and Drug Discovery: Structure-Based Drug Design and Structural Biology on GPCRs. In *G Protein-Coupled Receptors in Drug Discovery*, Leifert, W. R., Ed.; Humana Press: 2009, pp 51–66.

(6) Cohen, P. Protein Kinases - the Major Drug Targets of the Twenty-First Century? *Nat. Rev. Drug Discov.* **2002**, *1*, 309–315.

(7) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Deliver. Rev.* **1997**, *23*, 3–25.

(8) Lipinski, C. A. Lead- and Drug-like Compounds: the Rule-of-Five Revolution. *Drug Discov. Today: Technologies* **2004**, *1*, 337–341.

(9) Tetko, I. V.; Tanchuk, V. Y.; Villa, A. E. P. Prediction of *n*-Octanol/Water Partition Coefficients from PHYSPROP Database Using Artificial Neural Networkds and E-State Indices. *J. Chem. Inf. Model.* **2001**, *41*, 1407–1421.

(10) Hou, T. ADME Evaluation in Drug Discovery. 8. The Prediction of Human Intestinal Absorption by a Support Vector Machine. *J. Chem. Inf. Model.* **2007**, *47*, 2408–2415.

(11) Kortagere, S.; Chekmarev, D.; Welsh, W. J.; Ekins, S. New Predictive Models for Blood-Brain Barrier Permeability of Drug-like Molecules. *Pharm. Res.* **2008**, *25*, 1836–1845.

(12) Walters, W. P.; Murcko, M. A. Prediction of "Drug-likeness". *Adv. Drug Deliver. Rev.* **2002**, *54*, 255–271.

(13) Müller, K.-R.; Rätsch, G.; Sonnenburg, S.; Mika, S.; Grimm, M.; Heinrich, N. Classifying "Drug-likeness" with Kernel-Based Learning Methods. *J. Chem. Inf. Model.* **2005**, *45*, 249–253.

(14) Heikamp, K.; Bajorath, J. Support Vector Machines for Drug Discovery. *Expert Opin. Drug Dis.* **2014**, *9*, 93–104.

(15) Johnson, M. A.; Maggiora, G. M., *Concepts and Applications of Molecular Similarity*; Wiley: 1990.

(16) Ripphausen, P.; Nisius, B.; Peltason, L.; Bajorath, J. Quo Vadis, Virtual Screening? A Comprehensive Survey of Prospective Applications. *J. Med. Chem* **2010**, *53*, 8461–8467.

(17) Eckert, H.; Bajorath, J. Molecular Similarity Analysis in Virtual Screening: Foundations, Limitations and Novel Approaches. *Drug Discov. Today* **2007**, *12*, 225–233.

(18) Peltason, L.; Bajorath, J. SAR Index: Quantifying the Nature of Structure-Activity Relationships. *J. Med. Chem* **2007**, *50*, 5571–5578.

(19) Maggiora, G. M. On Outliers and Activity Cliffs - Why QSAR Often Disappoints. *J. Chem. Inf. Model.* **2006**, *46*, 1535.

(20) Guha, R.; Drie, J. H. V. Structure-Activity Landscape Index: Identifying and Quantifying Activity Cliffs. *J. Chem. Inf. Model.* **2008**, *48*, 646–658.

(21) Bajorath, J.; Peltason, L.; Wawer, M.; Guha, R.; Lajiness, M. S.; Drie, J. H. V. Navigating Structure-Activity Landscapes. *Drug Discov. Today* **2009**, *14*, 698–705.

(22) Peltason, L.; Iyer, P.; Bajorath, J. Rationalizing Three-Dimensional Activity Landscapes and the Influence of Molecular Representations on Landscape Topology and the Formation of Activity Cliffs. *J. Chem. Inf. Model.* **2010**, *50*, 1021–1033.

(23) Stumpfe, D.; Hu, Y.; Dimova, D.; Bajorath, J. Recent Progress in Understanding Activity Cliffs and Their Utility in Medicinal Chemistry. *J. Med. Chem* **2014**, *57*, 18–28.

(24) Stumpfe, D.; Bajorath, J. Methods for SAR Visualization. *RSC Adv.* **2012**, *2*, 369–378.

(25) Hu, Y.; Stumpfe, D.; Bajorath, J. Visualization of Activity Landscapes and Chemogenomics Data. *Mol. Inf.* **2013**, *32*, 954–963.

(26) Wassermann, A. M.; Wawer, M.; Bajorath, J. Activity Landscape Representations for Structure-Activity Relationship Analysis. *J. Med. Chem* **2010**, *53*, 8209–8223.

(27) Kruskal, J. B. Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika* **1964**, *29*, 1–27.

(28) Wawer, M.; Peltason, L.; Weskamp, N.; Teckentrup, A.; Bajorath, J. Structure-Activity Relationship Anatomy by Network-like Similarity Graphs and Local Structure-Activity Relationship Indices. *J. Med. Chem* **2008**, *51*, 6075–6084.

(29) Gupta-Ostermann, D.; Hu, Y.; Bajorath, J. Introducing the LASSO Graph for Compound Data Set Representation and Structure-Activity Relationship Analysis. *J. Med. Chem* **2012**, *55*, 5546–5553.

(30) Shanmugasundaram, V.; Maggiora, G. M. Characterizing Property and Activity Landscapes Using an Information-Theoretic Approach., 222nd American Chemical Society National Meeting, 2001.

(31) Klebe, G., *Wirkstoffdesign*, 2nd ed.; Springer Spektrum: 2009.

(32) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757–1768.

(33) Bolton, E. E.; Wang, Y.; Thiessen, P. A.; Bryant, S. H., PubChem: Integrated Platform of Small Molecules and Biological Activities. In *Annual Reports in Computational Chemistry*, Wheeler, R. A., Spellmeyer, D. C., Eds.; Elsevier: 2008; Chapter 12, pp 217–241.

(34) Wang, Y.; Suzek, T.; Zhang, J.; Wang, J.; He, S.; Cheng, T.; Shoemaker, B. A.; Gindulyte, A.; Bryant, S. H. PubChem BioAssay: 2014 update. *Nucleic Acids Res.* **2014**, *42*, D1075–D1082.

(35) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.

(36) Yung-Chi, C.; Prusoff, W. H. Relationship Between the Inhibition Constant K_I and the Concentration of Inhibitor Which Causes 50 Per Cent Inhibition I_{50} of an Enzymatic Reaction. *Biochem. Pharmacol.* **1973**, *22*, 3099–3108.

(37) Lazareno, S.; Birdsall, N. J. Estimation of Competitive Antagonist Affinity from Functional Inhibition Curves using the Gaddum, Schild and Cheng-Prusoff Equations. *Brit. J. Pharmacol.* **1993**, *109*, 1110–1119.

(38) Anderson, E.; Veith, G. D.; Weininger, D. *SMILES: A Line Notation and Computerized Interpreter for Chemical Structures*; tech. rep.; United States Environmental Protection Agency, 1987.

(39) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comp. Sci.* **1988**, *28*, 31–36.

(40) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *J. Chem. Theory Comput.* **1989**, *29*, 97–101.

(41) Weininger, D. SMILES. 3. DEPICT. Graphical Depiction of Chemical Structures. *J. Chem. Inf. Comp. Sci.* **1990**, *30*, 237–243.

(42) Kier, L. B. A Shape Index from Molecular Graphs. *Mol. Inf.* **1985**, *4*, 109–116.

(43) Randić, M. Novel Shape Descriptors for Molecular Graphs. *J. Chem. Inf. Comp. Sci.* **2001**, *41*, 607–613.

(44) MACCS Structural Keys., Accelrys: San Diego, CA, 2011.

(45) Molecular Operating Environment (MOE), 2013.08., Chemical Computing Group Inc., Montreal, Canada, 2013.

(46) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.

(47) Vogt, M.; Bajorath, J. Introduction of the Conditional Correlated Bernoulli Model of Similarity Value Distributions and its Application to the Prospective Prediction of Fingerprint Search Performance. *J. Chem. Inf. Model.* **2011**, *51*, 2496–2506.

(48) Gärtner, T.; Flach, P. A.; Wrobel, S., On Graph Kernels: Hardness Results and Efficient Alternatives. In *Proc. of the 16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop*, 2003, pp 129–143.

(49) Kashima, H.; Tsuda, K.; Inokuchi, A., Marginalized Kernels Between Labeled Graphs. In *Proc. of the 20th International Conference on Machine Learning*, 2003, pp 321–328.

(50) Leach, A. G.; Jones, H. D.; Cosgrove, D. A.; Kenny, P. W.; Ruston, L.; MacFaul, P.; Wood, J. M.; Colclough, N.; Law, B. Matched Molecular Pairs as a Guide in the Optimization of Pharmaceutical Properties; a Study of Aqueous Solubility, Plasma Protein Binding and Oral Exposure. *J. Med. Chem* **2006**, *49*, 6672–6682.

(51) Rogers, D. J.; Tanimoto, T. T. A Computer Program for Classifying Plants. *Science* **1960**, *132*, 1115–1118.

(52) Lavecchia, A. Machine-Learning Approaches in Drug Discovery: Methods and Applications. *Drug Discov. Today* **2015**, *20*, 318–331.

(53) Kohavi, R., A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proc. of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp 1137–1143.

(54) Mitchell, J. B. O. Machine Learning Methods in Chemoinformatics. *WIREs Comput. Mol. Sci.* **2014**, *4*, 468–481.

(55) Alpaydin, E., *Introduction to Machine Learning*, 2nd; MIT Press: 2010.

(56) Rojas, R., *Neural Networks - A Systematic Introduction*; Springer Berlin: 1996.

(57) Breiman, L.; Friedman, J.; Stone, C. J.; Olshen, R. A., *Classification and Regression Trees*; Chapmann and Hall: 1984.

(58) Mitchell, T., Decision Tree Learning. In *Machine Learning*, Munson, E. M., Ed.; McGraw Hill: 1997; Chapter 3, pp 52–80.

(59) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.

(60) Vapnik, V. N., *The Nature of Statistical Learning Theory*, 2nd ed.; Springer New York: 2000.

(61) Willett, P. Chemical Similarity Searching. *J. Chem. Inf. Comp. Sci.* **1998**, *38*, 983–996.

(62) Geppert, H.; Horváth, T.; Gärtner, T.; Wrobel, S.; Bajorath, J. Support-Vector-Machine-Based Ranking Significantly Improves the Effectiveness of Similarity Searching Using 2D Fingerprints and Multiple Reference Compounds. *J. Chem. Inf. Model.* **2008**, *48*, 742–746.

(63) Duda, R. O.; Hart, P. E.; Stork, D. G., *Pattern Classification*, 2nd ed.; Wiley-Interscience: 2000.

(64) Zhang, H., The Optimality of Naive Bayes. In *Proc. of the 17th International Florida Artificial Intelligence Research Society Conference*, 2004, pp 562–567.

(65) Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A. J.; Vapnik, V. N., Support Vector Regression Machines. In *Advances in Neural Information Processing Systems 9*, 1997, pp 155–161.

(66) Tsochantaridis, I.; Hofmann, T.; Joachims, T.; Altun, Y., Support Vector Machine Learning for Interdependent and Structured Output Spaces. In *Proc. of the 21st International Conference on Machine Learning*, 2004, pp 104–111.

(67) Cortes, C.; Vapnik, V. N. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297.

(68) Boser, B. E.; Guyon, I. M.; Vapnik, V. N., A Training Algorithm for Optimal Margin Classifiers. In *Proc. of the 5th Annual Workshop on Computational Learning Theory*, 1992, pp 144–152.

(69) Morik, K.; Brockhausen, P.; Joachims, T., Combining Statistical Learning with a Knowledge-Based Approach - A Case Study in Intensive Care Monitoring. In *Proc. of the 16th International Conference on Machine Learning*, 1999, pp 268–277.

(70) Ng, A. Support Vector Machines., In: CS229 Lecture Notes, `http://cs229.stanford.edu/notes/cs229-notes3.pdf`, accessed May 2015.

(71) Boyd, S.; Vandenberghe, L., *Convex Optimization*; Cambridge University Press: 2004.

(72) Mercer, J. Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations. *Philos. T. Roy. Soc. A* **1909**, *209*, 441–458.

(73) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph Kernels for Chemical Informatics. *Neural Networks* **2005**, *18*, 1093–1110.

(74) Mahé, P.; Ralaivola, L.; Stoven, V.; Vert, J.-P. The Pharmacophore Kernel for Virtual Screening with Support Vector Machines. *J. Chem. Inf. Model.* **2006**, *46*, 2003–2014.

(75) Jacob, L.; Vert, J.-P. Protein-Ligand Interaction Prediction: An Improved Chemogenomics Approach. *Bioinformatics* **2008**, *24*, 2149–2156.

(76) Wassermann, A. M.; Heikamp, K.; Bajorath, J. Potency-Directed Similarity Searching Using Support Vector Machines. *Chem. Biol. Drug Des.* **2011**, *77*, 30–38.

(77) Smola, A. J.; Schölkopf, B. A Tutorial on Support Vector Regression. *Stat. Comput.* **2004**, *14*, 199–222.

(78) Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* **2005**, *6*, 1453–1484.

(79) Joachims, T., Making Large-Scale Support Vector Machine Learning Practical. In *Advances in Kernel Methods*, Schölkopf, B., Burges, C. J. C., Smola, A. J., Eds.; MIT Press: 1999; Chapter 11, pp 169–184.

(80) Papadatos, G.; Alkarouri, M.; Gillet, V. J.; Willett, P. Lead Optimization Using Matched Molecular Pairs: Inclusion of Contextual Information for Enhanced Prediction of hERG Inhibition, Solubility, and Lipophilicity. *J. Chem. Inf. Model.* **2010**, *50*, 1872–1886.

(81) Sushko, Y.; Novotarskyi, S.; Körner, R.; Vogt, J.; Abdelaziz, A.; Tetko, I. V. Prediction-Driven Matched Molecular Pairs to Interpret QSARs and Aid the Molecular Optimization Process. *J. Cheminform.* **2014**, *6*.

(82) Carlsson, L.; Helgee, E. A.; Boyer, S. Interpretation of Nonlinear QSAR Models Applied to Ames Mutagenicity Data. *J. Chem. Inf. Model.* **2009**, *49*, 2551–2558.

(83) Mohr, J.; Jain, B.; Sutter, A.; Laak, A. T.; Steger-Hartmann, T.; Heinrich, N.; Obermayer, K. A Maximum Common Subgraph Kernel Method for Predicting the Chromosome Aberration Test. *J. Chem. Inf. Model.* **2010**, *50*, 1821–1838.

(84) Rosenbaum, L.; Hinselmann, G.; Jahn, A.; Zell, A. Interpreting Linear Support Vector Machine Models with Heat Map Molecule Coloring. *J. Cheminform.* **2011**, *3*.

(85) Martens, D.; Huysmans, J.; Setiono, R.; Vanthienen, J.; Baesens, B., Rule Extraction from Support Vector Machines: An Overview of Issues and Application in Credit Scoring. In *Rule Extraction from Support Vector Machines*, Diederich, J., Ed.; Springer Berlin Heidelberg: 2008; Chapter 2, pp 33–63.

(86) Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; Müller, K.-R. How to Explain Individual Classification Decisions. *J. Mach. Learn. Res.* **2010**, *11*, 1803–1831.

(87) Wassermann, A. M.; Haebel, P.; Weskamp, N.; Bajorath, J. SAR Matrices: Automated Extraction of Information-Rich SAR Tables from Large Compound Data Sets. *J. Chem. Inf. Model.* **2012**, *52*, 1769–1776.

(88) Gupta-Ostermann, D.; Balfer, J.; Bajorath, J. Hit Expansion from Screening Data Based upon Conditional Probabilities of Activity Derived from SAR Matrices. *Mol. Inf.* **2015**, *34*, 134–146.

(89) Bender, A.; Jenkins, J. L.; Glick, M.; Deng, Z.; Nettles, J. H.; Davies, J. W. "Bayes Affinity Fingerprints" Improve Retrieval Rates in Virtual Screening and Define Orthogonal Bioactivity Space: When are Multitarget Drugs a Feasible Concept? *J. Chem. Inf. Model.* **2006**, *46*, 2445–2456.

(90) Wassermann, A. M.; Lounkine, E.; Glick, M. Bioturbo Similarity Searching: Combining Chemical and Biological Similarity to Discover Structurally Diverse Bioactive Molecules. *J. Chem. Inf. Model.* **2013**, *53*, 692–703.

(91) Heikamp, K.; Hu, X.; Yan, A.; Bajorath, J. Prediction of Activity Cliffs Using Support Vector Machines. *J. Chem. Inf. Model.* **2012**, *52*, 2354–2365.

Appendix

Support Vector Machine Derivations

Linearly separable data

We can compute the margin as:

$$\rho(\mathbf{w}, b) = \min_{\{\mathbf{x}^{(i)}|y^{(i)}=+1\}} \frac{\mathbf{x}^{(i)} \cdot \mathbf{w}}{||\mathbf{w}||} - \max_{\{\mathbf{x}^{(i)}|y^{(i)}=-1\}} \frac{\mathbf{x}^{(i)} \cdot \mathbf{w}}{||\mathbf{w}||} \tag{56}$$

Considering the constraints in equation (19), the following hold:

$$\min_{\{\mathbf{x}^{(i)}|y^{(i)}=+1\}} \frac{\mathbf{x}^{(i)} \cdot \mathbf{w}}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||} \tag{57}$$

$$\max_{\{\mathbf{x}^{(i)}|y^{(i)}=-1\}} \frac{\mathbf{x}^{(i)} \cdot \mathbf{w}}{||\mathbf{w}||} = \frac{-1}{||\mathbf{w}||} \tag{58}$$

$$\rho(\mathbf{w}, b) = \frac{2}{||\mathbf{w}||} \tag{59}$$

Here, it becomes directly apparent that maximizing the margin can be done by minimizing $||\mathbf{w}||$. Usually, literature reports the minimization of $\frac{1}{2}\mathbf{w} \cdot \mathbf{w}$ for cosmetic reasons, which does not affect the solution [68].

The Lagrangian of the primal optimization problem for the classification SVM of linearly separable data is given by the primal optimization plus the linear constraints, for each of which a multiplier is added [71]. The constraints are first rearranged:

$$1 - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) \leq 0 \tag{60}$$

Then, the Lagrangian can be formulated and rearranged to arrive at Vapnik's formulation [60], which is also given in equation (20):

$$\Lambda(\mathbf{w}, b, \lambda) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + \sum_{i=1}^{n} \lambda^{(i)}[1 - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b)] \tag{61}$$

$$= \frac{1}{2}\mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{n} \lambda^{(i)}[y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) - 1] \tag{62}$$

The dual problem, which is maximized with respect to $\lambda^{(i)} \geq 0$, has to satisfy the KKT conditions [71]. They are given by the primal and dual constraint, the *complementary*

slackness in equation (65) which follows from the strong duality [71], and the fact that the gradient of the Lagrangian has to be zero at the solution.

$$1 - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) \leq 0 \tag{63}$$

$$\lambda^{(i)} \geq 0 \tag{64}$$

$$\lambda^{(i)}[1 - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b)] = 0 \tag{65}$$

$$\Delta\Lambda(\mathbf{w}, b, \lambda) = 0 \tag{66}$$

The partial derivatives of the Lagrangian are then given as:

$$\frac{\partial\Lambda(\mathbf{w}, b, \lambda)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} \tag{67}$$

$$\frac{\partial\Lambda(\mathbf{w}, b, \lambda)}{\partial b} = \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \tag{68}$$

$$\frac{\partial\Lambda(\mathbf{w}, b, \lambda)}{\partial \lambda^{(i)}} = \sum_{i=1}^{n} (y^{(i)} b - y^{(i)} \mathbf{x}^{(i)} \cdot \mathbf{w} + 1) \tag{69}$$

It follows from equation (67) and $\frac{\partial\Lambda(\mathbf{w}, b, \lambda)}{\partial \mathbf{w}} = 0$ that $\mathbf{w}$ can be expressed as:

$$\mathbf{w} = \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} \tag{70}$$

The Lagrangian can be rearranged, and by inserting equation (70), the final dual optimization problem is derived:

$$\Lambda(b, \lambda) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{n} \lambda^{(i)} [y^{(i)} (\mathbf{w} \cdot \mathbf{x}^{(i)} - b) - 1] \tag{71}$$

$$= \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} (\mathbf{w} \cdot \mathbf{x}^{(i)}) + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} + \sum_{i=1}^{n} \lambda^{(i)} \tag{72}$$

$$= \frac{1}{2} \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} \cdot \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \left(\sum_{j=1}^{n} \lambda^{(j)} y^{(j)} \mathbf{x}^{(j)} \cdot \mathbf{x}^{(i)} \right) + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} + \sum_{i=1}^{n} \lambda^{(i)} \tag{73}$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(j)} \cdot \mathbf{x}^{(i)}) + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} + \sum_{i=1}^{n} \lambda^{(i)} \tag{74}$$

$$= \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \tag{75}$$

We know that $\frac{\partial \Lambda(\mathbf{w},b,\lambda)}{\partial b} = \sum_{i=1}^{n} \lambda^{(i)} y^{(i)}$ has to be zero, and therefore, the last term can be omitted:

$$\Lambda(\lambda) = \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \tag{76}$$

Furthermore, the third KKT condition from equation (65) implies that either one of the following holds:

$$\lambda^{(i)} = 0 \tag{77}$$

$$1 - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) = 0 \tag{78}$$

Together with the primal constraints in equation (60), it follows that $\lambda^{(i)} \neq 0$ only where $y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) = 1$. This leads to the reduction of the summands in equation (70) to the support vectors, as shown in equation (21).

Since for all support vectors, $\mathbf{w} \cdot \mathbf{x}^{(i)} - b \in \{-1, +1\}$, b can be obtained using arbitrary $\mathbf{x}^{+}, \mathbf{x}^{-}$ from the set of support vectors with a positive and negative label, respectively:

$$\mathbf{w} \cdot \mathbf{x}^{+} - b = -(\mathbf{w} \cdot \mathbf{x}^{-} - b) \tag{79}$$

$$\Leftrightarrow \frac{1}{2}(\mathbf{w} \cdot \mathbf{x}^{+} + \mathbf{w} \cdot \mathbf{x}^{-}) = b \tag{80}$$

Noisy data

As a consequence of changing the primal problem formulation to equation (23) with constraints in equation (24), the dual problem changes to:

$$\begin{aligned} \Lambda(\mathbf{w}, b, \xi, \lambda, \nu) =& \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{n} \xi^{(i)} + \sum_{i=1}^{n} \lambda^{(i)} [1 - \xi^{(i)} - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b)] \\ & - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} \end{aligned} \tag{81}$$

Here, an additional set of dual variables ν is required to account for the constraints $\xi^{(i)} \geq 0$. From the new primal and dual problems, the KKT conditions are given as:

$$1 - \xi^{(i)} - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) \leq 0 \tag{82}$$

$$\xi^{(i)} \geq 0 \tag{83}$$

$$\lambda^{(i)} \geq 0 \tag{84}$$

$$\nu^{(i)} \geq 0 \tag{85}$$

$$\lambda^{(i)} [1 - \xi^{(i)} - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b)] = 0, \tag{86}$$

$$-\nu^{(i)} \xi^{(i)} = 0, \tag{87}$$

$$\Delta\Lambda(\mathbf{w}, b, \xi, \lambda, \nu) = 0 \tag{88}$$

In this case, the partial derivatives are:

$$\frac{\partial \Lambda(\mathbf{w}, b, \xi, \lambda, \nu)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \mathbf{x}^{(i)} \tag{89}$$

$$\frac{\partial \Lambda(\mathbf{w}, b, \xi, \lambda, \nu)}{\partial b} = \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \tag{90}$$

$$\frac{\partial \Lambda(\mathbf{w}, b, \xi, \lambda, \nu)}{\partial \xi^{(i)}} = C - \lambda^{(i)} - \nu^{(i)} \tag{91}$$

Interestingly, the partial derivatives with respect to $\mathbf{w}$ and b remain the same as in the linearly separable case, which means that equation (70) and equation (21) still hold.

Rearranging the Lagrangian in equation (81) and inserting equation (70) yields the following:

$$\begin{aligned}
\Lambda(b, \xi, \lambda, \nu) =& \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{n} \xi^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} [y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b) - 1 + \xi^{(i)} \\
& - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (92) \\
=& \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{n} \xi^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} (\mathbf{w} \cdot \mathbf{x}^{(i)}) + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \\
& + \sum_{i=1}^{n} \lambda^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} \xi^{(i)} - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (93) \\
=& \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) + C \sum_{i=1}^{n} \xi^{(i)} \\
& - \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \\
& + \sum_{i=1}^{n} \lambda^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} \xi^{(i)} - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (94) \\
=& \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \\
& + C \sum_{i=1}^{n} \xi^{(i)} + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} \xi^{(i)} - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (95)
\end{aligned}$$

Furthermore, rearranging equation (91), which has to be zero at the solution, gives two more equations:

$$\lambda^{(i)} = C - \nu^{(i)} \tag{96}$$

$$C = \lambda^{(i)} + \nu^{(i)} \tag{97}$$

Considering that equation (90) has to be zero and incorporating equation (97) into the Lagrangian, we arrive at:

$$\begin{aligned}
\Lambda(\lambda) =& \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \\
& + C \sum_{i=1}^{n} \xi^{(i)} + b \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} - \sum_{i=1}^{n} \lambda^{(i)} \xi^{(i)} - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (98) \\
=& \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \\
& + C \sum_{i=1}^{n} \xi^{(i)} - \sum_{i=1}^{n} (C - \nu^{(i)} \xi^{(i)} - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (99) \\
=& \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) \\
& + C \sum_{i=1}^{n} \xi^{(i)} - C \sum_{i=1}^{n} \xi^{(i)} + \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} && (100) \\
=& \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}) && (101)
\end{aligned}$$

Here it becomes apparent that the slack variables and their corresponding dual variables ν vanish from the problem.

Altogether, the same function as in the linearly separable case is derived, which has to be maximized subject to:

$$\sum_{i=1}^{n} \lambda^{(i)} y^{(i)} = 0 \tag{102}$$

$$0 \leq \lambda^{(i)} \leq C \tag{103}$$

Here, the box constraints on λ follow from equation (85) and equation (97).

Hence, the computation of $\mathbf{w}$ and the classification rule stays the same; only to compute b, $\mathbf{x}^+$ and $\mathbf{x}^-$ from equation (80) have to be chosen such that:

$$\mathbf{x}^+ \in \{\mathbf{x}^{(i)} | y^{(i)} = +1 \wedge \lambda^{(i)} < C\} \tag{104}$$

$$\mathbf{x}^- \in \{\mathbf{x}^{(i)} | y^{(i)} = -1 \wedge \lambda^{(i)} < C\} \tag{105}$$

This follows from equation (87), which tells us that either $\nu^{(i)} = 0$ or $\xi^{(i)} = 0$. Hence, it can be inferred that $\xi^{(i)} = 0$ where $\nu^{(i)} \neq 0$.

Nonlinear data

The mapping function only affects the constraints of the primal optimization problem, and thereby two of the KKT conditions in equation (82) and equation (86):

$$1 - \xi^{(i)} - y^{(i)}(\mathbf{w} \cdot \phi(\mathbf{x}^{(i)}) - b) \leq 0 \tag{106}$$

$$\lambda^{(i)}[1 - \xi^{(i)} - y^{(i)}(\mathbf{w} \cdot \phi(\mathbf{x}^{(i)}) - b)] = 0 \tag{107}$$

Consequently, the derivation of $\mathbf{w}$ and the rearranged Lagrangian $\Lambda(\lambda)$ change accordingly:

$$\mathbf{w} = \sum_{i=1}^{n} \lambda^{(i)} y^{(i)} \phi(\mathbf{x}^{(i)}) \tag{108}$$

$$\Lambda(\lambda) = \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} (\phi(\mathbf{x}^{(i)}) \cdot \phi(\mathbf{x}^{(j)})) \tag{109}$$

Using a kernel function $K(u, v)$, the Lagrangian, the derivation of b, and the decision function can be rewritten as:

$$\Lambda(\lambda) = \sum_{i=1}^{n} \lambda^{(i)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda^{(i)} \lambda^{(j)} y^{(i)} y^{(j)} K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \tag{110}$$

$$b = \frac{1}{2} \sum_{\text{support vectors}} \lambda^{(i)} y^{(i)} \left[K(\mathbf{x}^{(i)}, \mathbf{x}^{+}) + K(\mathbf{x}^{(i)}, \mathbf{x}^{-}) \right] \tag{111}$$

$$f(\mathbf{x}) = \sum_{\text{support vectors}} \lambda^{(i)} y^{(i)} K(\mathbf{x}^{(i)}, \mathbf{x}) - b \tag{112}$$

Imbalanced problems

Introducing two regularization terms C_+, C_- changes the primal optimization function shown in equation (36). The Lagrangian is then defined as:

$$\begin{aligned} \Lambda(\mathbf{w}, b, \xi, \lambda, \nu) = & \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C_+ \sum_{\{i | y^{(i)} = +1\}} \xi^{(i)} + C_- \sum_{\{i | y^{(i)} = -1\}} \xi^{(i)} \\ & + \sum_{i=1}^{n} \lambda^{(i)} [1 - \xi^{(i)} - y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} - b)] - \sum_{i=1}^{n} \nu^{(i)} \xi^{(i)} \end{aligned} \tag{113}$$

The KKT conditions remain the same as in the soft margin case, and only the partial derivative with respect to $\xi^{(i)}$ changes:

$$\frac{\partial \Lambda(\mathbf{w}, b, \xi, \lambda, \nu)}{\partial \xi^{(i)}} = \begin{cases} C_+ - \lambda^{(i)} - \nu^{(i)} & \{i | y^{(i)} = +1\} \\ C_- - \lambda^{(i)} - \nu^{(i)} & \{i | y^{(i)} = -1\} \end{cases} \tag{114}$$

Furthermore, equation (95) changes to:

$$\Lambda(b,\xi,\lambda,\nu) = \sum_{i=1}^{n}\lambda^{(i)} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda^{(i)}\lambda^{(j)}y^{(i)}y^{(j)}(\mathbf{x}^{(i)}\cdot\mathbf{x}^{(j)}) - \sum_{i=1}^{n}\nu^{(i)}\xi^{(i)}$$
$$+ C_+\sum_{\{i|y^{(i)}=+1\}}\xi^{(i)} + C_-\sum_{\{i|y^{(i)}=-1\}}\xi^{(i)} + b\sum_{i=1}^{n}\lambda^{(i)}y^{(i)} - \sum_{i=1}^{n}\lambda^{(i)}\xi^{(i)} \quad (115)$$

We can then insert $\lambda^{(i)}+\nu^{(i)}$ for C_+ and C_- analogously to arrive at the same formulation as in the soft margin case:

$$\begin{aligned}\Lambda(\lambda) =& \sum_{i=1}^{n}\lambda^{(i)} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda^{(i)}\lambda^{(j)}y^{(i)}y^{(j)}(\mathbf{x}^{(i)}\cdot\mathbf{x}^{(j)}) - \sum_{i=1}^{n}\nu^{(i)}\xi^{(i)}\\ &+ \sum_{\{i|y^{(i)}=+1\}}(\lambda^{(i)}+\nu^{(i)})\xi^{(i)} + \sum_{\{i|y^{(i)}=-1\}}(\lambda^{(i)}+\nu^{(i)})\xi^{(i)}\\ &+ b\sum_{i=1}^{n}\lambda^{(i)}y^{(i)} - \sum_{i=1}^{n}\lambda^{(i)}\xi^{(i)} \quad (116)\\ =& \sum_{i=1}^{n}\lambda^{(i)} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda^{(i)}\lambda^{(j)}y^{(i)}y^{(j)}(\mathbf{x}^{(i)}\cdot\mathbf{x}^{(j)}) - \sum_{i=1}^{n}\nu^{(i)}\xi^{(i)}\\ &+ \sum_{i=1}^{n}(\lambda^{(i)}+\nu^{(i)})\xi^{(i)} + b\sum_{i=1}^{n}\lambda^{(i)}y^{(i)} - \sum_{i=1}^{n}\lambda^{(i)}\xi^{(i)} \quad (117)\\ =& \sum_{i=1}^{n}\lambda^{(i)} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda^{(i)}\lambda^{(j)}y^{(i)}y^{(j)}(\mathbf{x}^{(i)}\cdot\mathbf{x}^{(j)}) \quad (118)\end{aligned}$$

Hence, only the maximization constraints and the choice of $\mathbf{x}^+, \mathbf{x}^-$ for the computation of b are altered:

$$0 \le \lambda^{(i)} \le C_+ \qquad i \in \{i|y^{(i)}=+1\} \quad (119)$$
$$0 \le \lambda^{(i)} \le C_- \qquad i \in \{i|y^{(i)}=-1\} \quad (120)$$
$$\mathbf{x}^+ \in \{\mathbf{x}^{(i)}|y^{(i)}=+1 \wedge \lambda^{(i)} < C_+\} \quad (121)$$
$$\mathbf{x}^- \in \{\mathbf{x}^{(i)}|y^{(i)}=-1 \wedge \lambda^{(i)} < C_-\} \quad (122)$$